

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A196 111

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

AFIT/CI/NR 88- 8

TITLE (and Subtitle)

NUMERICAL SOLUTIONS OF UNSTEADY
INVISCID TRANSONIC TURBINE CASCADE
FLOWS5. TYPE OF REPORT & PERIOD COVERED
MS THESIS

6. PERFORMING ORG. REPORT NUMBER

AUTHOR(s)

RICHARD McCREA MOORE

8. CONTRACT OR GRANT NUMBER(s)

PERFORMING ORGANIZATION NAME AND ADDRESS

FIT STUDENT AT: PURDUE UNIVERSITY

10. PROGRAM ELEMENT, PROJECT, TASK
AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
198813. NUMBER OF PAGES
264

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

AFIT/NR
Wright-Patterson AFB OH 45433-6583

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING
SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTED UNLIMITED: APPROVED FOR PUBLIC RELEASE

DTIC
ELECTE
AUG 03 1988
S D
C D

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

SAME AS REPORT

18. SUPPLEMENTARY NOTES

Approved for Public Release: IAW AFR 190-1

LYNN E. WOLAVER *Lynn Wolaver* 12 July 88
Dean for Research and Professional Development
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

ABSTRACT

Moore, Richard McCrea. Ph.D., Purdue University, May 1988. Numerical Solutions of Unsteady Inviscid Transonic Turbine Cascade Flows. Major Professor: Joe D. Hoffman, School of Mechanical Engineering.

↙ A numerical analysis has been developed to solve two-dimensional inviscid transonic turbine-type cascade flowfields. This analysis combines accuracy comparable to that of the numerical method of characteristics with the efficiency of finite difference methods. The MacCormack explicit finite difference method is used to solve the unsteady Euler equations. Steady solutions are calculated as asymptotic solutions in time. A conservation variable formulation of the Kentzer method has been developed in this investigation and is used to derive appropriate equations for the flowfield boundaries. The Kentzer method is based on characteristic theory, but uses a finite difference method, consistent with the method used at interior points, to integrate the appropriate boundary equations. A grid generator has been developed to create C-type grids around cascade blades using techniques similar to the Poisson equation grid generation techniques developed by Steger and Sorensen. Two different planar turbine-type cascades have been studied. The AACE II cascade blades are typical of the nozzle blades found in the first stator in a turbine. The GMA 400 cascade blades are typical of later turbine stator blades. Numerical studies were performed with maximum Mach

①

→ Theses, (Hoffman) ←

numbers in the flowfields ranging from 0.8 to 1.35. Numerical results are verified using experimentally measured blade surface static pressure data. A numerical method of characteristics cascade flow solver has been developed to provide a relative standard for numerical results. The MacCormack code and the characteristics code produce very similar results and both are in excellent agreement with the experimental results.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	



PURDUE UNIVERSITY

Graduate School

This is to certify that the thesis prepared

By Richard McCrea Moore

Entitled

Numerical Solutions of Unsteady Inviscid
Transonic Turbine Cascade Flows

Complies with University regulations and meets the standards of the Graduate School for
originality and quality

For the degree of Doctor of Philosophy

Signed by the final examining committee:

Joe D. Hoffman, chair
Robert V. Kuntz
H. Doyle Thompson
S. H. Fluke

Approved by the head of school or department:

April 28 1988 Wm. J. Judeman
School of Mechanical Engineering

This thesis ☐ is
☒ is not to be regarded as confidential

Joe D. Hoffman
Major professor

NUMERICAL SOLUTIONS OF
UNSTEADY INVISCID TRANSONIC
TURBINE CASCADE FLOWS

A Thesis
Submitted to the Faculty

of

Purdue University

by

Richard McCrea Moore

In Partial Fulfillment of the
Requirements for the degree

of

Doctor of Philosophy

May 1988

to
The United States of America
may it again become
One Nation Under God

ACKNOWLEDGMENTS

The development of the MacCormack flow solver, the numerical method of characteristics flow solver, and the C-grid generator described in this dissertation has been a sizeable undertaking, however, it was by no means the product of one man's efforts. First of all I thank my God who has "supplied all my needs according to his riches in glory in Christ Jesus." My wife, Jeannie, and my children, Christopher, Aaron, and Michelle deserve my deepest gratitude for their love, support and patience during my time at Purdue. I thank Dr. Joe Hoffman, my major professor, for his invaluable technical assistance, his excellent example as a teacher, and his constant willingness to lend a helping hand. I also thank Dr. Sanford Fleeter, Dr. Paul Kentzer, and Dr. H. Doyle Thompson for the good suggestions they provided as thesis committee members. The sponsor of my Ph.D. studies, the Department of Aeronautics of the United States Air Force Academy, made my desire to pursue a Ph.D. a reality. The experimental data supplied by the Allison Gas Turbine Division of the General Motors Corporation added greatly to the verification of the research results. Lastly, I thank my office partner, Richard F. Harwood, who shared in the high points and helped smooth out the rough times with his friendship and words of encouragement.

Richard McCrea Moore

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF SYMBOLS	x
ABSTRACT	xii
SECTION	
I. INTRODUCTION	1
I.1 OBJECTIVES	1
I.2 BACKGROUND	2
I.3 APPROACH	5
II. GOVERNING EQUATIONS AND COORDINATE TRANSFORMATIONS.....	6
II.1 GOVERNING EQUATIONS IN CONSERVATION VARIABLE FORM	6
II.2 GOVERNING EQUATIONS IN PRIMITIVE VARIABLE FORM ...	8
II.3 COORDINATE TRANSFORMATIONS AND THE STRONG CONSERVATION FORM	9
III. THE KENTZER METHOD	14
III.1 METHOD OF CHARACTERISTICS	15
III.2 THE KENTZER METHOD IN CONSERVATION VARIABLES ...	19
III.3 UNIT NORMAL VECTORS	22

IV. UNIT PROCESSES	25
IV.1 THE MACCORMACK METHOD	27
IV.2 INTERIOR POINT UNIT PROCESS	30
IV.3 BLADE SURFACE BOUNDARY POINT UNIT PROCESS	30
IV.4 TRAILING EDGE POINT UNIT PROCESS	33
IV.5 SUBSONIC INFLOW BOUNDARY POINT UNIT PROCESS	38
IV.6 EXIT BOUNDARY POINT UNIT PROCESSES	42
IV.7 GRID CUT POINTS	48
IV.8 PERIODIC BOUNDARY POINTS	49
V. GRID GENERATION	52
V.1 POISSON-TYPE CASCADE GRIDS	53
V.2 CONTROL OF GRID GEOMETRY NEAR BOUNDARIES	56
VI. SOLUTION PROCEDURE	61
VI.1 INITIAL-VALUE SURFACE	62
VI.2 TIME STEPPING PROCEDURE	63
VII. RESULTS	71
VII.1 NUMERICAL TEST CASES	71
VII.2 EXPLICIT ARTIFICIAL DISSIPATION	114
VII.3 TRAILING EDGE POINT	116
VIII. CONCLUSIONS	123
LIST OF REFERENCES	125
APPENDICES	
APPENDIX A: GOVERNING EQUATIONS AND COORDINATE TRANSFORMATIONS	128
APPENDIX B: METHOD OF CHARACTERISTICS	145
APPENDIX C: CONSERVATION VARIABLE COMPATIBILITY EQUATIONS	176
APPENDIX D: UNIT PROCESSES	184
APPENDIX E: GRID GENERATION	222
APPENDIX F: EXPLICIT ARTIFICIAL DISSIPATION	249
VITA	264

LIST OF TABLES

Table	Page
7-1. Cascade numerical test cases.	72

LIST OF FIGURES

Figure	Page
3-1. Stream surfaces and corresponding pathline.	17
3-2. Mach conoid, wave surface and corresponding waveline.	18
3-3. Unit vectors, \hat{b} , \hat{c} , \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 on three boundaries.	24
4-1. Trailing edge point unit vectors and flow directions.	35
5-1. C-type grid for the AACE II cascade blade.	55
6-1. The two-dimensional MacCormack method convex hull.	65
7-1. High density AACE II cascade grid. (165x13)	75
7-2. Low density AACE II cascade grid. (83x7)	76
7-3. High density GMA 400 cascade grid. (195x11)	77
7-4. Test Case 1. Blade surface normalized static pressure. AACE II cascade, baseline case.	79
7-5. Test Case 1. Static pressure contours.	81
7-6. Test Case 1. Mach number contours.	82
7-7. Test Case 2. Blade surface normalized static pressure. AACE II cascade, method of characteristics.	84
7-8. Test Case 2. Static pressure contours.	86
7-9. Test Case 2. Mach number contours.	87

Figure	Page
7-10. Test Case 3. Blade surface normalized static pressure. AAACE II cascade, coarse grid.	88
7-11. Test Case 3. Static pressure contours.	90
7-12. Test Case 3. Mach number contours.	91
7-13. Test Case 4. Blade surface normalized static pressure. AAACE II cascade, method of characteristics, coarse grid.	93
7-14. Test Case 4. Static pressure contours.	94
7-15. Test Case 4. Mach number contours.	95
7-16. Test Case 5. Blade surface normalized static pressure. AAACE II cascade, subsonic flowfield.	98
7-17. Test Case 5. Static pressure contours.	99
7-18. Test Case 5. Mach number contours.	100
7-19. Test Case 6. Blade surface normalized static pressure. AAACE II cascade, high transonic flowfield.	101
7-20. Test Case 6. Static pressure contours.	103
7-21. Test Case 6. Mach number contours.	104
7-22. Test Case 7. Blade surface normalized static pressure. GMA 400 cascade, subsonic flowfield.	107
7-23. Test Case 7. Static pressure contours.	108
7-24. Test Case 7. Mach number contours.	109
7-25. Test Case 8. Blade surface normalized static pressure. GMA 400 cascade, high subsonic flowfield.	111
7-26. Test Case 8. Static pressure contours.	112
7-27. Test Case 8. Mach number contours.	113
7-28. Viscous flow past a rounded trailing edge.	117
7-29. Inviscid flow past a rounded trailing edge.	118

Appendix Figure	Page
B-1. Stream surfaces and corresponding pathline.	147
B-2. Mach conoid, wave surface and corresponding waveline.	148
B-3. Convex hull for the numerical method of characteristics.	153
B-4. Pathline, four wavelines, and two solution surfaces.	155
D-1. Unit vectors \hat{b} , \hat{c} , \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 on three boundaries.	191
D-2. Trailing edge point unit vectors and flow directions.	199
E-1. H-type cascade blade grid.	226
E-2. O-type cascade blade grid.	228
E-3. C-type cascade blade grid.	229
E-4. C-type grid physical and computational boundaries.	230
E-5. C-type grid generated by the Laplace equation.	233
E-6. C-type grid generated by the Poisson equation.	234
E-7. Slope control of grid lines crossing the periodic boundaries.	241
E-8. Grid boundaries broken into 15 segments.	245
E-9. Cascade with nonzero stagger.	247
E-10. Grid system for a detuned cascade.	248

LIST OF SYMBOLS

a	speed of sound
B	MacCormack method backward difference operator
\hat{b}	unit vector perpendicular to a physical boundary
\mathcal{C}	space derivative terms in the conservation variable continuity equation
\hat{c}	unit vector tangent to a physical boundary
c_2, c_4	property independent artificial dissipation scaling factors
\mathcal{E}	space derivative terms in the conservation variable energy equation
F	MacCormack method forward difference operator
I	inverse Jacobian of the coordinate transformation
i	grid point counter in the ξ direction in computational space
\hat{i}	x-direction unit vector
J	Jacobian of the coordinate transformation
j	grid point counter in the η direction in computational space
\hat{j}	y-direction unit vector
\vec{M}	space derivative terms in the conservation variable vector momentum equation
n	time level counter
\hat{n}	wave surface unit normal vector, $\hat{n}_1, \hat{n}_2, \hat{n}_3$

P, Q	Poisson equation source term used in grid generation
P	pressure
$pc\xi_2, pc\eta_2$	property dependent second-order artificial dissipation scaling factors
$pc\xi_4, pc\eta_4$	property dependent fourth-order artificial dissipation scaling factors
P_0	inflow total pressure
Q, E, F	vectors containing Euler equation solution and space derivative terms
$\bar{Q}, \bar{E}, \bar{F}$	vectors containing strong conservation form Euler equation solution and space derivative terms
s	curve in physical space following a specified grid line
$sf\xi_2, sf\eta_2$	grid dependent second-order artificial dissipation scaling factors
$sf\xi_4, sf\eta_4$	grid dependent fourth-order artificial dissipation scaling factors
t	time
T_0	inflow total temperature
u, v	cartesian velocity components
U, V	contravariant velocity components
\bar{V}	velocity vector in physical space
V^2	square of the magnitude of the velocity vector
x, y	cartesian coordinates
γ	ratio of specific heats or cascade stagger angle
ρ	density
ξ, η	transformed coordinates
θ	inflow angle

ABSTRACT

Moore, Richard McCrea. Ph.D., Purdue University, May 1988. Numerical Solutions of Unsteady Inviscid Transonic Turbine Cascade Flows. Major Professor: Joe D. Hoffman, School of Mechanical Engineering.

A numerical analysis has been developed to solve two-dimensional inviscid transonic turbine-type cascade flowfields. This analysis combines accuracy comparable to that of the numerical method of characteristics with the efficiency of finite difference methods. The MacCormack explicit finite difference method is used to solve the unsteady Euler equations. Steady solutions are calculated as asymptotic solutions in time. A conservation variable formulation of the Kentzer method has been developed in this investigation and is used to derive appropriate equations for the flowfield boundaries. The Kentzer method is based on characteristic theory, but uses a finite difference method, consistent with the method used at interior points, to integrate the appropriate boundary equations. A grid generator has been developed to create C-type grids around cascade blades using techniques similar to the Poisson equation grid generation techniques developed by Steger and Sorensen. Two different planar turbine-type cascades have been studied. The AACE II cascade blades are typical of the nozzle blades found in the first stator in a turbine. The GMA 400 cascade blades are typical of later turbine stator blades. Numerical studies were performed with maximum Mach

numbers in the flowfields ranging from 0.8 to 1.35. Numerical results are verified using experimentally measured blade surface static pressure data. A numerical method of characteristics cascade flow solver has been developed to provide a relative standard for numerical results. The MacCormack code and the characteristics code produce very similar results and both are in excellent agreement with the experimental results.

SECTION I

INTRODUCTION

The analysis of cascade flowfields has many applications in the fields of mechanical, aeronautical, and astronautical engineering. The goal of the present investigation is to develop a cascade flowfield analysis technique which combines accuracy comparable to the numerical method of characteristics with the efficiency of finite difference techniques.

The specific objectives of the present investigation are presented in Section I.1. Section I.2 presents the background for this research. The approach chosen to achieve the stated objectives is outlined in Section I.3.

I.1 OBJECTIVES

The objectives of the research effort described in this dissertation are:

1. To develop a robust, general grid generator which will create body fitted C-type grids around two-dimensional turbine nozzle blades using automatic Poisson-type elliptic grid generation techniques.
2. To develop an accurate and efficient flow solver which will numerically calculate unsteady, inviscid, subsonic/transonic flowfields through two-

dimensional turbine-type cascades using the MacCormack explicit finite difference method and the Kentzer method applied in conservation variables at the boundaries.

3. To validate the grid generator and the flow solver using high quality experimental data and numerical results.

I.2 BACKGROUND

Any numerical technique requires some form of discretization of the flowfield. In order to achieve second-order accuracy, which is desirable for most engineering applications, the flowfield and boundaries are normally fit with a body-fitted grid. This grid is transformed into an equally spaced, orthogonal computational grid. For the current investigation, these requirements are achieved using an automatic Poisson-type grid solver to fit a C-type grid around a typical blade in the cascade being studied. As with any elliptic grid solver, all boundary locations are specified as Dirichlet boundary conditions. A technique developed in the present investigation is used, in conjunction with one developed by Steger and Sorenson [1], to automatically control grid point distribution in the interior of the flowfield. In addition to allowing a high level of accuracy, this grid generation technique provides significant generality since grids can be generated for two-dimensional cascades of arbitrary camber angle, solidity, and blade shape.

Many flowfields are trisonic, that is they contain subsonic, sonic, and supersonic flow regions. When evaluated in their steady state form, the Euler equations are elliptic for subsonic flows, parabolic for sonic flows, and

hyperbolic for supersonic flows. Numerically solving steady trisonic flows using these equations requires careful matching of different numerical techniques for each flow region. Therefore, a more efficient and more generalized approach involves considering the entire flowfield simultaneously using the unsteady Euler equations. In their time dependent form, the Euler equations are hyperbolic for all three flow regimes. Obviously, this approach provides unsteady flow solutions directly. Steady flow solutions are calculated as asymptotic solutions in time. This more general time dependent approach is used for the present investigation.

The MacCormack explicit finite difference method [2] is used to integrate the Euler equations numerically. Finite difference techniques are generally more computationally efficient than other types of numerical techniques. The MacCormack method is a particularly good method due to its efficiency and its second-order accuracy in both time and space.

The Euler equations can be solved using either conservation variables or primitive variables. Both approaches have advantages. For example, the highly accurate numerical method of characteristics is developed using the primitive variable form of the governing equations. However, for many finite difference methods, accuracy, robustness, and efficiency are enhanced by solving the Euler equations in conservation variables. Conservation variables are employed in the present investigation.

A technique proposed by Kentzer [3] permits accurate and efficient calculation of boundary point properties. This technique uses characteristic

theory to determine the appropriate governing equations at each type of flowfield boundary. Then, for the sake of computational efficiency, Kentzer suggests using a finite difference method to solve the resulting set of equations. In the present investigation, the same finite difference method used at interior points, the MacCormack explicit method, is also used for boundary point calculations. Marcum and Hoffman [4] have implemented the MacCormack method in conjunction with the Kentzer method boundary treatment, using the primitive variable form of the Euler equations.

Marcum [5] proposed a conservation variable version of the Kentzer method. This conservation variable development of the Kentzer method is based on primitive variable characteristic theory, wherein the compatibility equations are transformed to a conservation variable form. In the present investigation the conservation variable Kentzer method is developed. Specifically, the conservation variable Kentzer method is used to implement boundary conditions for the blade surface boundary, the trailing edge point, the subsonic inflow boundary, and the subsonic exit boundary. Prior to the present development of the conservation variable Kentzer method, the analyst who chose to use the Kentzer method was required to solve the entire flowfield in primitive variables or to inconsistently solve interior points using conservation variables and boundary points using primitive variables.

I.3 APPROACH

The concepts discussed in Section I.2 are combined to produce an accurate and efficient numerical solution technique for inviscid turbine-type cascade flowfields. The resulting MacCormack flow solver is evaluated by calculating flowfield solutions for two significantly different cascade geometries and by calculating solutions over the full range of subsonic and transonic Mach numbers representative of turbine flowfields. These solutions are validated through comparison with high quality experimental data and with numerical results obtained by the numerical method of characteristics.

Subsonic and transonic blade surface static pressure measurements are available for the AACE II [6] and the GMA 400 [7] turbine type cascades. These results provide an absolute test for the results of the MacCormack code.

The numerical method of characteristics matches numerics to the physics it models more accurately than other numerical methods. Thus, it provides an excellent standard of accuracy for other numerical methods. An efficient inverse marching numerical method of characteristics inviscid cascade flow solver is developed, as part of the current investigation, to provide a relative comparison for the MacCormack flow solver.

SECTION II

GOVERNING EQUATIONS AND COORDINATE TRANSFORMATIONS

This section presents the governing equations for the present investigation in a variety of conservation variable and primitive variable forms. One conservation variable form is also shown transformed into strong conservation form. An expanded discussion of the various forms of the governing equations and the transformation is provided in Appendix A.

II.1 GOVERNING EQUATIONS IN CONSERVATION VARIABLE FORM

The equations applicable to this research effort are the continuity equation, the vector momentum equation, the energy equation, and the thermal and caloric equations of state.

$$(\rho)_t + \nabla \cdot (\rho \vec{V}) = 0 \quad (2.1)$$

$$(\rho \vec{V})_t + \nabla \cdot (\rho \vec{V} \vec{V}) + \nabla P = 0 \quad (2.2)$$

$$(\rho e)_t + \nabla \cdot [\vec{V}(\rho e + P)] = 0 \quad (2.3)$$

$$T = \frac{P}{\rho R} \quad (2.4)$$

$$a = \left[\frac{\gamma P}{\rho} \right]^{1/2} \quad (2.5)$$

These equations are based on the following assumptions:

1. continuum flow,
2. inviscid flow,
3. no body forces,
4. no heat conduction,
5. simple thermodynamic system, and
6. thermally and calorically perfect gas.

Equations (2.1) to (2.3) can be presented in the following short hand notation.

$$(\rho)_t = \mathcal{C} \quad (2.6)$$

$$(\rho \vec{V})_t = \vec{\mathcal{M}} \quad (2.7)$$

$$(\rho e)_t = \mathcal{E} \quad (2.8)$$

where \mathcal{C} , $\vec{\mathcal{M}}$ and \mathcal{E} contain the space derivatives appearing in the continuity, vector momentum, and energy equations, respectively. Written in vector notation, \mathcal{C} , $\vec{\mathcal{M}}$ and \mathcal{E} are:

$$\mathcal{C} = -\nabla \cdot (\rho \vec{V}) \quad (2.9)$$

$$\vec{M} = -\nabla \cdot (\rho \vec{V} \vec{V}) - \nabla P \quad (2.10)$$

$$\mathcal{E} = -\nabla \cdot [\vec{V}(\rho e + P)] \quad (2.11)$$

The matrix form of the governing equations is:

$$Q_t + E_x + F_y = 0 \quad (2.12)$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad E = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (\rho e + P)u \end{bmatrix} \quad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (\rho e + P)v \end{bmatrix}$$

II.2 GOVERNING EQUATIONS IN PRIMITIVE VARIABLE FORM

The conservation variable form of the continuity equation, equation (2.1), is expanded to yield the primitive variable form:

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) = 0 \quad (2.13)$$

Expanding equation (2.2) into primitive variables and using the continuity equation, equation (2.13), to simplify the result yields the following primitive variable form of the vector momentum equation:

$$\rho \frac{D\vec{V}}{Dt} + \nabla P = 0 \quad (2.14)$$

The energy equation is used in two different primitive variable forms in later sections and appendices. Appendix A shows the expansion of equation (2.3) into both of the following primitive variable forms:

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (2.15)$$

$$\frac{1}{\gamma-1} \frac{DP}{Dt} + \frac{V^2}{2} \frac{D\rho}{Dt} + \rho \frac{D \left[\frac{V^2}{2} \right]}{Dt} + \frac{\gamma P}{\gamma-1} (\nabla \cdot \vec{V}) + \rho \frac{V^2}{2} (\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P = 0 \quad (2.16)$$

II.3 COORDINATE TRANSFORMATIONS AND THE STRONG CONSERVATION FORM

Coordinate transformations are often utilized in numerical computations to improve solution accuracy. This requires two major steps: the transformation of the grid covering the physical domain of interest to a uniform orthogonal computational grid, and the transformation of the governing equations from the physical domain to the computational domain. The grid transformation is addressed in Section V and in Appendix E. The transformation of the governing equations is presented in this section, with more detail afforded in Appendix A.

II.3.1 COORDINATE TRANSFORMATIONS

Since neither the physical nor the computational grid is time dependent in the present investigation, derivatives with respect to time are unaffected by the transformation. Therefore, only space transformations are considered.

The physical and transformed spaces are related through a one-to-one correspondence. That is, each point in the physical domain, (x,y) , is related to exactly one point in the computational domain, (ξ,η) , and vice versa. Therefore, the computational space is described by:

$$\xi = \xi(x,y) \quad (2.17)$$

$$\eta = \eta(x,y) \quad (2.18)$$

or conversely,

$$x = x(\xi,\eta) \quad (2.19)$$

$$y = y(\xi,\eta) \quad (2.20)$$

Partial derivatives of the generic variable, $f = f(x,y)$, are transformed using the partial differentiation chain rule:

$$f_x = f_\xi \xi_x + f_\eta \eta_x \quad (2.21)$$

$$f_y = f_\xi \xi_y + f_\eta \eta_y \quad (2.22)$$

The terms ξ_x , ξ_y , η_x , and η_y are the transformation metrics. These metrics are calculated using the following equations:

$$\xi_x = J y_\eta \quad (2.23)$$

$$\eta_x = -J y_\xi \quad (2.24)$$

$$\xi_y = -J x_\eta \quad (2.25)$$

$$\eta_y = J x_\xi \quad (2.26)$$

where J is the Jacobian of the transformation. By definition J is given by the following determinant:

$$J = \begin{vmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{vmatrix} \quad (2.27)$$

This determinant is equivalent to the reciprocal of the Jacobian of the inverse transformation, I . In other words, $J = I^{-1}$, where

$$I = \begin{vmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{vmatrix} \quad (2.28)$$

Physically, the Jacobian represents a stretching function which relates a differential volume in physical space to that same volume in the transformed space.

Since the computational grid in the transformed space is orthogonal and equally spaced, all the inverse metrics $(x_\xi, x_\eta, y_\xi, y_\eta)$ can be calculated numerically to a high-order of accuracy. As Thompson [8] points out, these metrics should be numerically approximated in the same manner as the dependent variables are approximated in the governing equations. For the MacCormack method, used in the present investigation, the metrics should be forward differenced for the predictor calculations and backward differenced for the corrector calculations. This issue is discussed further in Appendix A.

II.3.2 GOVERNING EQUATIONS IN STRONG CONSERVATION FORM

Using the partial differentiation chain rule to transform the space derivatives in equation (2.12) as shown in equations (2.21) and (2.22) results in

the following system of equations in weak conservation form.

$$Q_t + \xi_x E_\xi + \eta_x E_\eta + \xi_y F_\xi + \eta_y F_\eta = 0 \quad (2.29)$$

Using a technique developed by Viviand [9] and Vinokur [10], equation (2.29) can be recast into the strong conservation form:

$$[IQ]_t + [I(E\xi_x + F\xi_y)]_\xi + [I(E\eta_x + F\eta_y)]_\eta = 0 \quad (2.30)$$

The development of equation (2.30) is presented in Appendix A.

II.3.3 CONTRAVARIANT VELOCITIES

Equation (2.30) contains several groupings of velocity products of the following form:

$$U = \xi_x u + \xi_y v \quad (2.31)$$

$$V = \eta_x u + \eta_y v \quad (2.32)$$

These groupings are called contravariant velocities. Contravariant velocities represent transformed velocity components in the directions of the computational coordinates. Specifically, U represents the velocity component in the ξ direction, and V represents the velocity component in the η direction.

Substituting these contravariant velocities into the transformed governing equations improves computational efficiency. Thus, the strong conservation law governing equations, with the contravariant velocities incorporated, are:

$$\bar{Q}_t + \bar{E}_\xi + \bar{F}_\eta = 0 \quad (2.33)$$

where

$$\bar{Q} = I \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad \bar{E} = \begin{bmatrix} I(\rho U) \\ I(\rho u U + \xi_x P) \\ I(\rho v U + \xi_y P) \\ I[(\rho e + P)U] \end{bmatrix} \quad \bar{F} = \begin{bmatrix} I(\rho V) \\ I(\rho u V + \eta_x P) \\ I(\rho v V + \eta_y P) \\ I[(\rho e + P)V] \end{bmatrix}$$

In this research program, equation (2.33) is solved using the MacCormack explicit finite difference method to calculate properties at all points in the interior of the flowfield. The Kentzer method yields similar sets of equations applicable at the various boundary points.

SECTION III

THE KENTZER METHOD

Many techniques exist for calculating fluid properties at the boundaries of a flowfield. At a boundary the physics are described by the applicable boundary condition equations and the governing differential equations of motion. Thus, the number of equations exceeds the number of solution variables and the problem is over specified. Therefore, to incorporate the boundary conditions, some boundary point solution techniques ignore one or more of the governing equations and, therefore, some important physics.

The Kentzer method [3] combines the boundary conditions and the governing differential equations of motion into a new set of equations which includes all the applicable physics yet does not over specify the problem. The Kentzer method is based on characteristic theory. Characteristic theory is described in Appendix B and summarized in this section. Detailed developments of the method of characteristics have been published by Rusanov [11], Zucrow and Hoffman [12], and Hoffman [13].

Characteristic theory uses the primitive variable form of the governing equations. Therefore, up to the present, the Kentzer method has been applied

using primitive variables. However, a conservation variable form of the Kentzer method, which was first proposed by Marcum [5], is developed and applied in the present investigation. This variation of the Kentzer method is described in this section. The conservation variable Kentzer method is applied to several boundary conditions in Section IV.

III.1 METHOD OF CHARACTERISTICS

The method of characteristics identifies the physical paths of propagation of information through a flowfield, which are the pathline and the wavesurfaces. This identification process is developed in Appendix B. The results of that development are summarized below.

The governing equations for the present investigation are presented in Section II. These equations form a set of hyperbolic partial differential equations of the first order. In primitive variables they are:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{V} = 0 \quad (3.1)$$

$$\frac{D\vec{V}}{Dt} + \frac{\nabla P}{\rho} = 0 \quad (3.2)$$

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (3.3)$$

The goal of the method of characteristics is to form linear combinations of this set of equations to obtain an equivalent set of compatibility equations. A compatibility equation is an interior operator which has one less independent variable, contains derivatives only in the corresponding characteristic surface,

and is valid only in that surface. A characteristic surface is a surface in the solution space on which the governing partial differential equations may be combined linearly to form a compatibility equation.

For unsteady two-dimensional flow there are three independent variables: (x, y, t) . Therefore, the characteristic surfaces are surfaces in three-dimensional space. The pathline is the intersection of all the stream surfaces containing the point in space which is under consideration. The waveline is the line of contact between a wave surface and the Mach conoid. The Mach conoid is the envelope of all wave surfaces which contain the given point in space which is under consideration. Two stream surfaces and the corresponding pathline are illustrated in Figure 3-1. A Mach conoid, a wave surface, and the corresponding waveline are illustrated in Figure 3-2.

There are two types of compatibility equations corresponding to the two types of characteristic surfaces: the pathline equation and the waveline equation. The pathline equation is the energy equation in primitive variable form, equation (3.3), which is repeated here for convenience.

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (3.4)$$

$\frac{D(\quad)}{Dt}$ is the substantial derivative, which is the directional derivative along the pathline. As demonstrated in Appendix B, the waveline equation is not as simple. The waveline equation is formed as a linear combination of equations (3.1) to (3.3), which reduces to:

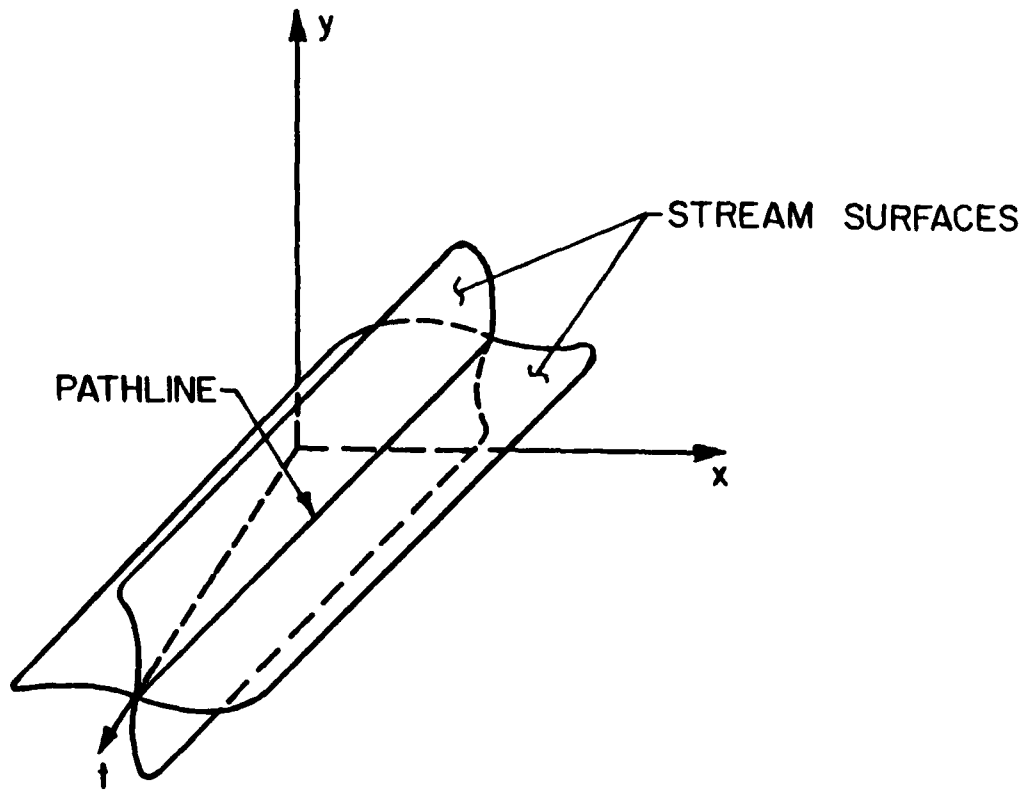


Figure 3-1. Stream surfaces and corresponding pathline.

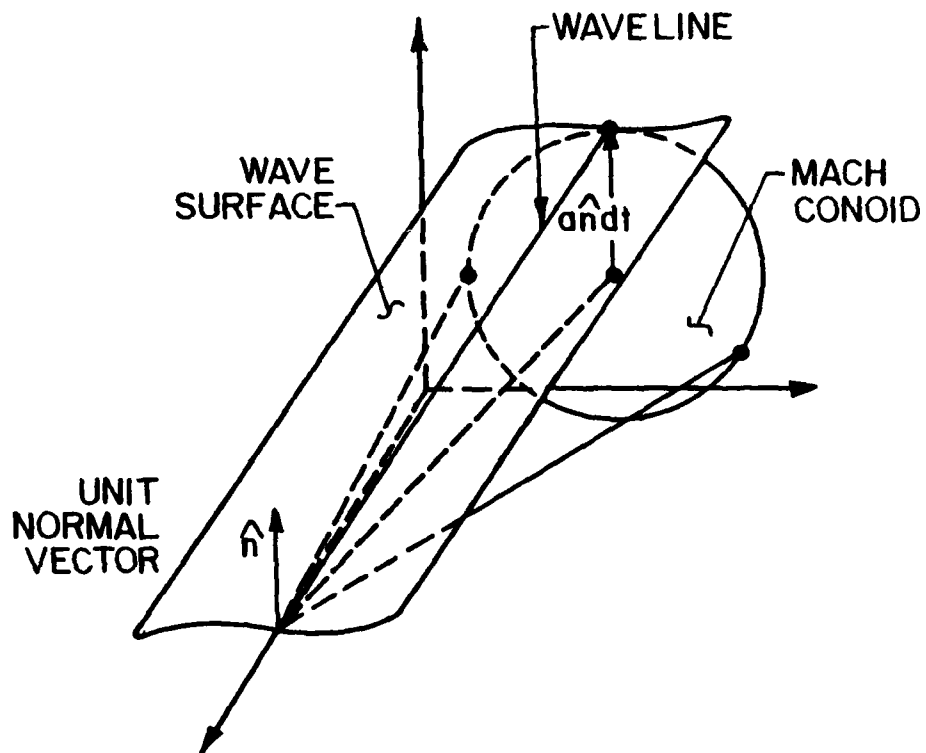


Figure 3-2. Mach conoid, wave surface and corresponding waveline.

$$\frac{\partial P}{\partial t} - \rho a \hat{n} \cdot \frac{\partial \vec{V}}{\partial t} + \rho a^2 \left[\nabla \cdot \vec{V} - \hat{n} \cdot (\hat{n} \cdot \nabla) \vec{V} \right] = 0 \quad (3.5)$$

where $\frac{\partial(\)}{\partial t}$ is the directional derivative along the waveline. The term in brackets contains derivatives in the corresponding wave surface which are called cross derivatives.

The pathline equation, equation (3.4), is valid on the pathline. The waveline equation, equation (3.5), is valid on the wave surface corresponding to the wave surface unit normal vector, \hat{n} . There are an infinite number of choices for the wave surface unit normal vector \hat{n} corresponding to the infinite number of wave surfaces at a point. However, only four compatibility equations can be included as a set of independent equations for unsteady two-dimensional flow. The pathline equation must be included since it is the only compatibility equation which contains a derivative of density. Consequently, three and only three waveline equations can be used independently for unsteady two-dimensional flow.

III.2 THE KENTZER METHOD IN CONSERVATION VARIABLES

As shown above, characteristic theory uses the primitive variable form of the governing differential equations of motion to derive the pathline and waveline equations. Therefore, the Kentzer method has historically been applied in primitive variables. However, a conservation variable form of the Kentzer method is developed and applied in the present investigation.

The conservation variable form of the governing equations written in vector notation, equations (2.6) to (2.8), are repeated here for convenience:

$$(\rho)_t = \mathcal{E} \quad (3.6)$$

$$(\rho \vec{V})_t = \vec{\mathcal{M}} \quad (3.7)$$

$$(\rho e)_t = \mathcal{E} \quad (3.8)$$

where

$$\mathcal{E} = -\nabla \cdot (\rho \vec{V}) \quad (3.9)$$

$$\vec{\mathcal{M}} = -\nabla \cdot (\rho \vec{V} \vec{V}) - \nabla P \quad (3.10)$$

$$\mathcal{E} = -\nabla \cdot [\vec{V}(\rho e + P)] \quad (3.11)$$

A linear combination of these conservation variable governing equations produces the conservation variable form of the pathline equation, which reduces to:

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{V} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{V} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (3.12)$$

In a similar manner, a linear combination of the conservation variable form of the governing equations produces the conservation variable form of the waveline equation, which reduces to:

$$\begin{aligned}
& \left[\frac{V^2}{2} + \frac{a\hat{n} \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a\hat{n}}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\
& = \left[\frac{V^2}{2} + \frac{a\hat{n} \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a\hat{n}}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (3.13)
\end{aligned}$$

Appendix C demonstrates the mathematical equivalence of the primitive variable and the conservation variable compatibility equations. While these two sets of equations are mathematically equivalent, only the primitive variable compatibility equations can be used to construct a numerical method of characteristics, because only the primitive variable equations are made up solely of derivatives which lie in characteristic surfaces. However, since the Kentzer method uses finite differences to solve the boundary point equations, either set of compatibility equations is acceptable.

By applying the Kentzer method to these conservation variable compatibility equations, the appropriate set of conservation variable equations is derived for each type of boundary being considered. At boundary points, the governing equations, equations (3.6), (3.7), and (3.8), are replaced by the pathline equation, equation (3.12), and the waveline equation, equation (3.13), applied in three independent wave surfaces. Depending upon what type of boundary is being considered, one or more of these compatibility equations are replaced by boundary conditions. Since each compatibility equation contains all the physics described by the governing equations, all the appropriate physics is included in the equations produced by the Kentzer method.

At interior points, the same set of compatibility equations could be used, but they reduce to the original set of governing differential equations of

motion. It is only when one or more boundary conditions replace one or more of the compatibility equations that the resulting set of applicable equations differs from the original governing differential equations of motion.

At every boundary point, the resulting equations are solved by the same numerical method used at interior points. For the present investigation, that method is the MacCormack explicit forward-backward finite difference method.

III.3 UNIT NORMAL VECTORS

In order to implement the boundary point equations, a local coordinate system is established at the boundary point. The local boundary coordinates are defined by \hat{b} and \hat{c} , where \hat{b} is the unit vector normal to the boundary and \hat{c} is the unit vector tangent to the boundary.

At any given point, an infinite number of wave surfaces exist which could be chosen for application of the waveline equation. Since only three waveline equations are considered at each boundary point, the wave surface unit normal vectors, \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , are chosen such that they are equally distributed and aligned with the boundary at the point being considered. The \hat{n}_1 wave surface unit normal vector is chosen to be equal to the boundary unit vector, \hat{b} . Thus, in terms of the unit vectors \hat{b} and \hat{c} , the three wave surface unit normal vectors are defined in the following manner:

$$\hat{n}_1 = \hat{b} \quad (3.14)$$

$$\hat{n}_2 = -\frac{1}{2} \hat{b} + \frac{\sqrt{3}}{2} \hat{c} \quad (3.15)$$

$$\hat{n}_3 = -\frac{1}{2} \hat{b} - \frac{\sqrt{3}}{2} \hat{c} \quad (3.16)$$

All five of these unit vectors, \hat{b} , \hat{c} , \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , are shown at three boundary point locations on a C-type grid in Figure 3-3.

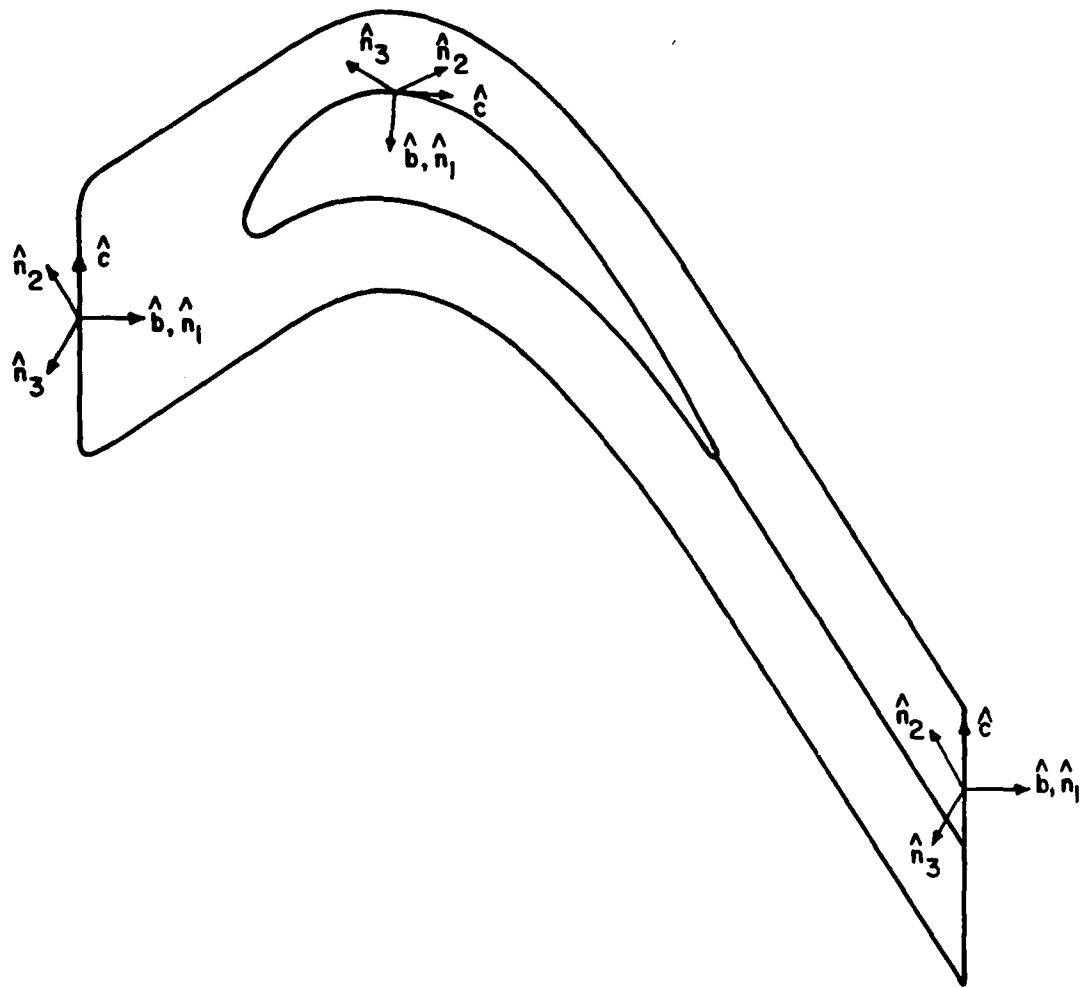


Figure 3-3. Unit vectors \hat{b} , \hat{c} , \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 on three boundaries.

SECTION IV

UNIT PROCESSES

Several unit processes are employed in the computation of cascade flowfields. Each of the following types of flowfield points must be considered:

1. interior points,
2. solid wall boundary points,
3. trailing edge point,
4. subsonic inlet boundary points,
5. subsonic exit boundary points, and
6. supersonic exit boundary points.

All periodic boundary points and grid cut points lie in one of the regions listed above. Therefore, no additional sets of equations are required to compute these points. However, because of their location on the computational grid, they do require special handling of indices.

This section summarizes the MacCormack method and the results of the Kentzer method unit process development for each of the flow regions listed above. The Kentzer method is discussed in Section III. Appendix D presents

each of the unit process developments in detail.

The equations derived in this section are presented in terms of physical domain (x,y,t) derivatives. Prior to their implementation, these equations are transformed to computational space. At the interior points the governing equations are transformed into the strong conservation form as shown in equation (2.33). At the boundaries the transformed equations consists of the same groupings of space derivatives transformed into strong conservation form, however, these derivative terms have coefficients which are not affected by the transformation. Therefore, the boundary point equations are in a weak conservation form.

At interior points the Euler equations apply. Those equations are repeated here for convenience:

$$(\rho)_t = \mathcal{C} \quad (4.1)$$

$$(\rho \vec{V})_t = \vec{\mathcal{M}} \quad (4.2)$$

$$(\rho e)_t = \mathcal{E} \quad (4.3)$$

where \mathcal{C} , $\vec{\mathcal{M}}$, and \mathcal{E} contain the space derivatives appearing in the continuity, vector momentum, and energy equations, respectively. Written in vector notation, \mathcal{C} , $\vec{\mathcal{M}}$, and \mathcal{E} are:

$$\mathcal{C} = -\nabla \cdot (\rho \vec{V}) \quad (4.4)$$

$$\vec{\mathcal{M}} = -\nabla \cdot (\rho \vec{V} \vec{V}) - \nabla P \quad (4.5)$$

$$\mathcal{E} = -\nabla \cdot [\vec{V}(\rho e + P)] \quad (4.6)$$

At all boundary points, the Kentzer method is used to determine the appropriate set of equations to be solved. At all interior and all boundary points, the space derivative terms \mathcal{C} , \vec{M} , and \mathcal{E} appear. In addition, at all interior and all boundary points, the MacCormack explicit finite difference method is used to discretize the appropriate equations.

IV.1 THE MACCORMACK METHOD

The MacCormack explicit finite difference method [2] is used in this investigation to calculate the flowfield solution. This predictor-corrector method is second-order accurate in time and space.

The transformed governing equations in matrix form are presented in Section II.3.3 and are repeated here for convenience:

$$\bar{Q}_t + \bar{E}_\xi + \bar{F}_\eta = 0 \quad (4.7)$$

where

$$\bar{Q} = I \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad \bar{E} = \begin{bmatrix} I(\rho U) \\ I(\rho u U + \xi_x P) \\ I(\rho v U + \xi_y P) \\ I[(\rho e + P)U] \end{bmatrix} \quad \bar{F} = \begin{bmatrix} I(\rho V) \\ I(\rho u V + \eta_x P) \\ I(\rho v V + \eta_y P) \\ I[(\rho e + P)V] \end{bmatrix}$$

The two steps of the MacCormack method can be summarized, for the present investigation, by the following equations. The predictor space derivatives are all computed using current time level property values:

$$\bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left[\mathbf{F} \left(\bar{E}_\xi \right)_{i,j}^n + \mathbf{F} \left(\bar{F}_\eta \right)_{i,j}^n \right] \quad (4.8)$$

The corrector space derivatives are all computed using forward time level property values:

$$\bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left[\mathbf{B} \left(\bar{E}_\xi \right)_{i,j}^{n+1} + \mathbf{B} \left(\bar{F}_\eta \right)_{i,j}^{n+1} \right] \quad (4.9)$$

To achieve second order accuracy in time and space, the results of these two steps are averaged:

$$\bar{Q}_{i,j}^{n+1} = \frac{1}{2} \left[\bar{Q}_{i,j}^{n+1} + \bar{Q}_{i,j}^{n+1} \right] \quad (4.10)$$

Efficiency is improved by combining equations (4.9) and (4.10) in the following manner:

$$\bar{Q}_{i,j}^{n+1} = \frac{1}{2} \left[\bar{Q}_{i,j}^{n+1} + \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left\{ \mathbf{B} \left(\bar{E}_\xi \right)_{i,j}^{n+1} + \mathbf{B} \left(\bar{F}_\eta \right)_{i,j}^{n+1} \right\} \right] \quad (4.11)$$

As shown in equations (4.8), (4.9), and (4.11), $\mathbf{F}(\)$ are first-order forward-difference operators and $\mathbf{B}(\)$ are first-order backward-difference operators. Those operators are:

$$\mathbf{F}(\bar{E}_\xi)_{i,j}^n = \frac{(\bar{E}_{i+1,j}^n - \bar{E}_{i,j}^n)}{\Delta \xi}$$

$$\mathbf{F}(\bar{F}_\eta)_{i,j}^n = \frac{(\bar{F}_{i,j+1}^n - \bar{F}_{i,j}^n)}{\Delta \eta}$$

$$\mathbf{B}(\bar{E}_\xi)_{i,j}^{n+1} = \frac{(\bar{E}_{i,j}^{n+1} - \bar{E}_{i-1,j}^{n+1})}{\Delta\xi}$$

$$\mathbf{B}(\bar{F}_\eta)_{i,j}^{n+1} = \frac{(\bar{F}_{i,j}^{n+1} - \bar{F}_{i,j-1}^{n+1})}{\Delta\eta}$$

By incorporating the forward-difference operators, the MacCormack predictor equation, equation (4.8), applied to the transformed governing equations in matrix form becomes:

$$\bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left[\frac{\bar{E}_{i+1,j}^n - \bar{E}_{i,j}^n}{\Delta\xi} + \frac{\bar{F}_{i,j+1}^n - \bar{F}_{i,j}^n}{\Delta\eta} \right] \quad (4.12)$$

Similarly, by incorporating the backward-difference operators, the MacCormack corrector equation, equation (4.11), applied to the transformed governing equations in matrix form becomes:

$$\bar{Q}_{i,j}^{n+1} = 0.5 \left[\bar{Q}_{i,j}^{n+1} + \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left\{ \frac{\bar{E}_{i,j}^{n+1} - \bar{E}_{i-1,j}^{n+1}}{\Delta\xi} + \frac{\bar{F}_{i,j}^{n+1} - \bar{F}_{i,j-1}^{n+1}}{\Delta\eta} \right\} \right] \quad (4.13)$$

For the present investigation, all grid spacings on the computational grid are unity (i.e., $\Delta\xi = \Delta\eta = 1.0$).

At each of the boundaries, one of the forward- or the backward-difference operations requires flow properties and transformation metrics outside the flowfield to approximate flow property partial derivatives at the boundaries. Therefore, first-order or second-order extrapolation is used to approximate these values outside boundaries. Then the MacCormack method, as described above, is applied to the appropriate boundary point equations to evaluate boundary property values.

IV.2 INTERIOR POINT UNIT PROCESS

At interior points, the MacCormack method is applied directly to the transformed governing equations in the strong conservation-law form, equation (2.33). The resulting finite difference equations, equations (4.12) and (4.13), are solved at each interior point to determine the solution at the new time level.

IV.3 BLADE SURFACE BOUNDARY POINT UNIT PROCESS

In a two-dimensional cascade flow, the surface of the blade forms the only solid wall boundary. The boundary condition applicable at a free slip solid boundary point is that the velocity normal to the boundary is zero:

$$\hat{\mathbf{b}} \cdot \vec{\mathbf{V}} = 0 \quad (4.14)$$

To determine the applicable equations at the blade surface boundary point, the governing equations, equations (4.1) to (4.3), are replaced by compatibility equations, equations (3.12) and (3.13). The waveline equation, equation (3.13), is applied in three wave surfaces corresponding to the wave surface unit normal vectors, $\hat{\mathbf{n}}_1$, $\hat{\mathbf{n}}_2$, and $\hat{\mathbf{n}}_3$, given by equations (3.14) to (3.16). Since the boundary condition, equation (4.14), must be incorporated, the waveline equation corresponding to vector $\hat{\mathbf{n}}_1$ (which was intentionally placed outside the flowfield) is replaced by the boundary condition. This situation is illustrated in Figure 3-3. The applicable equations at a solid wall boundary point are the boundary condition, equation (4.14), the pathline equation,

equation (3.12), and the waveline equation, equation (3.13), applied in the remaining two wave surfaces corresponding to wave surface unit normal vectors \hat{n}_2 and \hat{n}_3 . Thus,

$$\hat{b} \cdot \vec{V} = 0 \quad (4.15)$$

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{V} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{V} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (4.16)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.17)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.18)$$

Equations (4.15) to (4.18) comprise an appropriate set of equations for the solid wall boundary point. However, these equations are unnecessarily complicated. Equations (4.16) to (4.18) each contain more than one time derivative. These four equations can be rearranged into a form which is simpler and more computationally efficient to solve by a finite difference

method. That rearrangement results in the following set of equations:

$$\rho_t = \mathcal{E} + \left[\frac{1}{2a} \right] \hat{\mathbf{b}} \cdot \vec{\mathcal{M}} \quad (4.19)$$

$$(\rho u)_t = \mathcal{M}_i + \left[\frac{u}{2a} - b_i \right] (\hat{\mathbf{b}} \cdot \vec{\mathcal{M}}) \quad (4.20)$$

$$\rho v = \left[\frac{c_j}{c_i} \right] \rho u \quad (4.21)$$

$$(\rho e)_t = \mathcal{E} + \left[\frac{V^2}{4a} + \frac{a}{2(\gamma-1)} \right] (\hat{\mathbf{b}} \cdot \vec{\mathcal{M}}) \quad (4.22)$$

Equations (4.19) to (4.22) are solved at each solid blade surface boundary point to determine the solution at the new time level.

The MacCormack method backward corrector calculations require predicted property values one row of grid points inside the solid boundary. This is achieved by extrapolating predicted values from the flowfield. Both linear and quadratic extrapolations have been used at the blade surface during the current research effort. With either approach some explicit smoothing is required to stabilize the solution. Using linear extrapolation produces good results. When quadratic extrapolation is used the smoothing requirement is increased to such an extent that some significant flow features are smeared out. Therefore, linear extrapolation is used at the blade surface boundary points in the present investigation.

IV.4 TRAILING EDGE POINT UNIT PROCESS

In the inviscid cascade flowfield, the Kutta condition is enforced at the trailing edge of the cascade blade. Specifically, the flows leaving the two sides of the blade surface at the trailing edge are forced to flow parallel to each other in a direction such that the static pressure is equal on the two sides of the trailing edge point.

In the present investigation, the requirements listed above are enforced by installing a small imaginary solid wall segment which is hinged at the trailing edge point. The flow on each side of the hinged wall segment is solved independently, in a manner similar to the technique described in Section IV.3 for the solid blade surface point. By forcing the flow on both sides of the wall segment to follow the wall, the flows on the two sides of the trailing edge are forced to be parallel to each other.

After solving for the flow properties on each side of the hinged wall segment independently, the pressure difference between the two sides is compared. If the difference is not equal to zero, to within a specified tolerance, a zero finding secant method is used to correct the wall orientation (flow angle) toward the angle where the pressures will be equal. This procedure is iterated until the pressure difference between the pressure and suction sides of the blade, at the trailing edge, is approximately zero.

As mentioned above, the flow on each side of the imaginary hinged wall segment is solved in a manner similar to the grid points which lie on the remainder of the blade surface. However, because the imaginary wall segment

is not stationary, but instead is allowed to pivot, the local coordinate systems, set up on each side of the wall at the trailing edge points, also pivot. This is illustrated in Figure 4-1.

Throughout the trailing edge point unit process derivation, unless otherwise specified, the local coordinate system unit vectors, \hat{b} and \hat{c} , are defined with respect to their orientation at the forward time level, that is, the time level of the predicted and corrected properties. Therefore, unless otherwise specified, during the iterative solution process, \hat{b} is aligned perpendicular to, and \hat{c} is aligned parallel to the hinged wall segment at the angle the wall is assumed to be at, at the forward time level, during that iteration.

The boundary condition applicable at the trailing edge point's imaginary hinged wall segment is that the velocity normal to the wall segment, at the hinge, at the new time step, is zero.

$$\hat{b} \cdot \vec{V} = 0 \quad (4.23)$$

To determine the applicable equations at the trailing edge point, the governing equations, equations (4.1) to (4.3), are replaced with compatibility equations, equations (3.12) and (3.13). The waveline equation, equation (3.13), is applied in three wave surfaces corresponding to the wave surface unit normal vectors, \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (3.14) to (3.16). Since $\hat{n}_1 = \hat{b}$, the waveline equation corresponding to vector \hat{n}_1 extends beyond the hinged wall segment and therefore, does not affect the flow on the side of the wall under consideration. As at the blade surface boundary condition, the \hat{n}_1 waveline is intentionally placed outside of the flow and is replaced by the

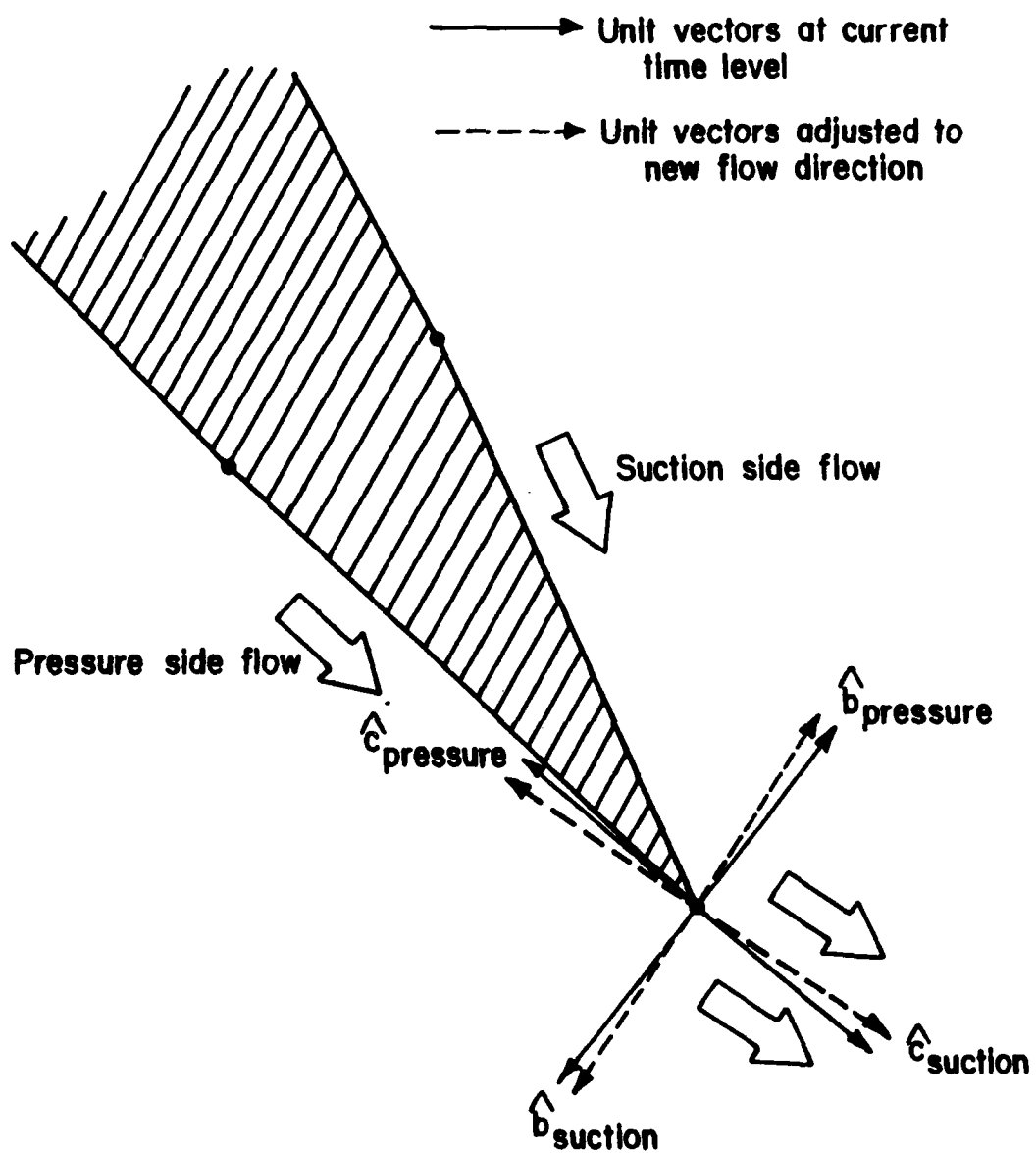


Figure 4-1. Trailing edge point unit vectors and flow directions.

boundary condition, equation (4.23). Thus, the applicable equations at the trailing edge point are the boundary condition, equation (4.23), the pathline equation, equation (3.12), and the waveline equation, equation (3.13), applied in the remaining two wave surfaces corresponding to wave surface unit normal vectors \hat{n}_2 and \hat{n}_3 . Thus,

$$\hat{b} \cdot \vec{V} = 0 \quad (4.24)$$

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{V} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{V} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (4.25)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.26)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.27)$$

Equations (4.24) to (4.27) comprise an appropriate set of equations for the flows on the two sides of the hinged wall segment (trailing edge point). However, these equations are unnecessarily complicated. Equations (4.25) to (4.27) each contain more than one time derivative. These four equations can

be rearranged into a form which is simpler and more computationally efficient to solve by a finite difference method. That rearrangement results in the following set of equations:

$$\rho_t = \mathcal{E} + \frac{\hat{\mathbf{b}} \cdot [(\rho \vec{\mathbf{V}})_t - \vec{\mathcal{M}}]}{[\hat{\mathbf{b}} \cdot \vec{\mathbf{V}} - 2a]} \quad (4.28)$$

$$(\rho u)_t = \frac{\mathcal{M}_i + g \left[-\mathcal{M}_j + \left(\frac{c_j}{c_i} \right)_t (\rho u) \right]}{1 - g \left(\frac{c_j}{c_i} \right)} \quad (4.29)$$

$$\rho v = \left(\frac{c_j}{c_i} \right) \rho u \quad (4.30)$$

$$(\rho e)_t = \mathcal{E} + \left[\vec{\mathbf{V}} + \left\{ \frac{\frac{V^2}{2} - \frac{a^2}{\gamma-1}}{2a - \hat{\mathbf{b}} \cdot \vec{\mathbf{V}}} \right\} \hat{\mathbf{b}} \right] \cdot [(\rho \vec{\mathbf{V}})_t - \vec{\mathcal{M}}] \quad (4.31)$$

where $g = (u - 2ab_i)/(v - 2ab_j)$. In the denominator of equation (4.29), the $\hat{\mathbf{c}}$ vector is parallel to the hinged wall segment at the current time level for the predictor calculations, and is parallel to the hinged wall segment at its assumed position at the forward time level for the corrector calculations. For both steps, the temporal derivative of the ratio of the $\hat{\mathbf{c}}$ components is equal to the ratio at the forward time level, minus the ratio at the current time level, all divided by the time step.

Equations (4.28) to (4.31) are solved on each side of the imaginary hinged wall segment located at the blade trailing edge to advance the solution in

time. The solution process is iterated, changing the flow angle at each iteration until the pressures on the two sides of the trailing edge are equal to within a specified tolerance. For the results presented in Section VII, a tolerance of 10^{-6} was used to evaluate convergence of the pressure difference normalized by the inflow total pressure.

The same set of equations must be solved at the predictor and the corrector steps of the MacCormack method. Therefore, the flow angle, and the corresponding forward time level definitions of \hat{b} and \hat{c} , must be the same for both steps. This constrains the iteration sequence to include both the predictor and corrector steps. Therefore, all points on the C-grid except the trailing edge point are predicted, next the trailing edge point is iteratively predicted and corrected, then all remaining points are corrected.

IV.5 SUBSONIC INFLOW BOUNDARY POINT UNIT PROCESS

When the component of the fluid velocity perpendicular to the inflow boundary is less than the local speed of sound, the fluid properties at that boundary are dependent upon both upstream and downstream phenomena. If the fluid velocity were zero, the fluid properties would be influenced from all directions equally. When the velocity is not negligible, a greater influence will be felt from the upstream direction. From a characteristic perspective, this means that most of the base of the Mach cone (domain of dependence) lies outside the computational domain at the subsonic inflow boundary.

At an inflow boundary, one to four boundary conditions can be set. It would only be appropriate to set four boundary conditions if the flow component crossing the inflow boundary were supersonic, and therefore, the flow properties were totally dependent upon the upstream conditions. Setting three boundary conditions implies that a majority, but not all of the flow influences are coming from the upstream direction. For the present investigation three boundary conditions are set at the inflow boundary.

The boundary conditions chosen for the subsonic inflow boundary are those which reflect the properties of the flow which are most likely to be known upstream of a turbine blade row. Specifically, the stagnation pressure, P_0 , the stagnation temperature, T_0 , and the flow angle, θ , are specified as boundary conditions at the inflow boundary.

To determine the applicable equations at the subsonic inflow boundary, the governing equations, equations (4.1) to (4.3), are replaced by compatibility equations, equations (3.12) and (3.13). The waveline equation, equation (3.13), is applied in three wave surfaces corresponding to the wave surface unit normal vectors, \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (3.14) to (3.16). At the inflow boundary the unit vector \hat{b} is oriented perpendicular to the inflow boundary and pointing in. Thus, the wavelines corresponding to the \hat{n}_2 and \hat{n}_3 vectors and the pathline, which are all outside the computational domain, are replaced by the three boundary conditions. Only the waveline corresponding to the \hat{n}_1 unit vector is placed inside the computational domain. This situation is illustrated in Figure 3-3. Thus, the applicable equations at the subsonic inflow boundary are the three properties, P_0 , T_0 , and θ , and the

waveline equation applied along the \hat{n}_1 unit vector.

$$P_0 = P_{0_{inflow}} \quad (4.32)$$

$$T_0 = T_{0_{inflow}} \quad (4.33)$$

$$\theta = \theta_{inflow} \quad (4.34)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a\hat{n}_1 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a\hat{n}_1}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a\hat{n}_1 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a\hat{n}_1}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.35)$$

Equations (4.32) to (4.35) comprise an appropriate set of equations for the inflow boundary. In previous unit process derivations, multiple compatibility equations made it possible to use linear algebra to simplify the applicable equations prior to implementation. Since only one compatibility equation is used at the subsonic inflow boundary, it must be computed without simplification. This means that all the temporal derivatives in equation (4.35) must be evaluated simultaneously. This is achieved through iteration.

The Mach number, M , is the iteration variable in the subsonic inflow boundary point unit process. Knowing an estimated forward time Mach number and the boundary conditions given by equations (4.32) to (4.34), several additional properties can be computed. In this unit process, the density, ρ , the x-direction and y-direction components of momentum, ρu and

ρv , and the energy, ρe , are needed at the forward time for the iterative process. Therefore, in terms of known flow properties and the Mach number:

$$\rho = \frac{P_0}{RT_0} \left[\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right]^{\frac{1}{\gamma-1}} \quad (4.36)$$

$$\rho u = P_0 M \left[\frac{\gamma}{RT_0 \left(1 + \tan^2(\theta) \right) \left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma+1}{\gamma-1}}} \right]^{\frac{1}{2}} \quad (4.37)$$

$$\rho v = \rho u \tan(\theta) \quad (4.38)$$

$$\rho e = P_0 \left[\frac{\gamma}{\gamma-1} - \frac{1}{1 + \frac{\gamma-1}{2} M^2} \right] \left[\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right]^{\frac{1}{\gamma-1}} \quad (4.39)$$

The iterative process used to find property values at inflow boundary points is described next. For both predictor and corrector steps, the iterative procedure is initiated by guessing the forward time Mach number at the inflow grid point being considered. For the first time step, the Mach number guess is calculated from property values on the initial-value surface. Subsequently the initial Mach number guess is calculated from current time level property values. The guessed Mach number is used in equations (4.36) to (4.39) to calculate guessed forward time values of ρ , ρu , ρv , and ρe .

Equation (4.35) is rearranged to solve for the temporal derivative of energy:

$$(\rho e)_t = \mathcal{E} - \left[\frac{V^2}{2} + \frac{a\hat{b} \cdot \vec{V}}{\gamma-1} \right] (\rho_t - \mathcal{E}) + \left[\vec{V} + \frac{a\hat{b}}{\gamma-1} \right] \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (4.40)$$

Temporal derivatives of density and momentum are calculated numerically using the current and guessed forward time values of these properties. Using these temporal derivatives and transformed space derivatives, equation (4.40) is solved for a calculated value of ρe . If the calculated and guessed values of ρe are equal, to within a small tolerance, the Mach number guess was correct and the four guessed forward time property values are the predicted (or corrected) property values at the new time step. If the calculated and guessed values of ρe are not approximately equal, a zero finding secant method is used to home in on the correct Mach number. For the results presented in Section VII, a tolerance of 10^{-6} was used to evaluate convergence of the energy difference normalized by the value of the energy at the current time level.

For the predictor step, forward differenced space derivatives are required. Therefore, quadratic extrapolation is used to approximate property values outside the computational domain.

IV.6 EXIT BOUNDARY POINT UNIT PROCESSES

Because the component of the flow velocity which crosses the exit boundary can be either subsonic or supersonic, each exit boundary point must be checked, at each time step, to determine which unit process is appropriate. At the exit boundary, the unit vector \hat{b} is placed perpendicular to the boundary, pointing outward, as illustrated in Figure 3-3. Therefore, at each exit

boundary point the quantity $\hat{b} \cdot \vec{V}$ is calculated and compared to the local speed of sound, a , to determine whether the boundary point should be handled as a supersonic or a subsonic exit boundary point.

When the component of the fluid velocity perpendicular to the exit boundary is less than the local speed of sound, the fluid properties at that boundary are dependent upon both upstream and downstream phenomena. Therefore, at least one boundary condition must be applied to the solution procedure. This situation is discussed in Section IV.6.1.

When the component of the fluid velocity perpendicular to the exit boundary is greater than or equal to the local speed of sound, the fluid mechanics at the exit are only dependent upon upstream conditions. From a characteristic perspective, this means that the base of the Mach cone (domain of dependence) lies entirely within the computational domain. In this case the interior point unit process is appropriate at an exit boundary point. This situation is discussed in Section IV.6.2.

On a C-type grid, the exit boundary is a single straight line with a length equal to the cascade blade spacing. However, the two ends of the computational grid ($\xi = 1 = \text{constant}$ and $\xi = \xi_{\max} = \text{constant}$ grid lines) make up the exit boundary. This distinction does not affect the derivation of applicable equations at the exit boundaries. However, the computations on these two boundaries must be implemented in a slightly different manner. Specifically, the numerical approximation of space derivatives in the ξ direction requires property value extrapolation beyond the exit at the

predictor step for the $\xi = \xi_{\max}$ boundary, and at the corrector step for the $\xi = 1$ boundary. In both cases, quadratic extrapolation is used.

IV.6.1 SUBSONIC EXIT BOUNDARY POINT UNIT PROCESS

The boundary condition applied at a subsonic exit boundary point is that the exit static pressure is known:

$$P = P_{\text{exit}} \quad (4.41)$$

Because pressure is not one of the four solution variables being computed at each grid point, the solution procedure at the subsonic exit is iterative.

To determine the applicable equations at the subsonic exit boundary point, the governing equations, equations (4.1) to (4.3), are replaced by compatibility equations, equations (3.12) and (3.13). The waveline equation, equation (3.13), is applied in three wave surfaces corresponding to the wave surface unit normal vectors, \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (3.14) to (3.16). Since the boundary condition, equation (4.41), must be incorporated, the waveline equation corresponding to vector \hat{n}_1 (which was intentionally placed outside the flowfield) is replaced by the boundary condition. This situation is illustrated in Figure 3-3. The applicable equations at a subsonic exit boundary point are the boundary condition, equation (4.41), the pathline equation, equation (3.12), and the waveline equation, equation (3.13), applied in the remaining two wave surfaces corresponding to wave surface unit normal vectors \hat{n}_2 and \hat{n}_3 :

$$P = P_{\text{exit}} \quad (4.42)$$

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{V} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{V} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (4.43)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.44)$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (4.45)$$

Equations (4.42) to (4.45) comprise an appropriate set of equations for the subsonic exit boundary point. However, these equations are unnecessarily complicated. Equations (4.43) to (4.45) each contain more than one time derivative. Equations (4.43) to (4.45) can be rearranged into a form which is simpler and more computationally efficient to solve by a finite difference method. That rearrangement results in the following set of equations:

$$P = P_{\text{exit}} \quad (4.46)$$

$$\rho_t = \mathcal{E} + \frac{(\rho e)_t - \mathcal{E}}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2\mathbf{a}\hat{\mathbf{b}}\cdot\vec{\mathbf{V}}} \quad (4.47)$$

$$(\rho u)_t = \mathcal{M}_i + \frac{[(\rho e)_t - \mathcal{E}] [u - 2ab_i]}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2\mathbf{a}\hat{\mathbf{b}}\cdot\vec{\mathbf{V}}} \quad (4.48)$$

$$(\rho v)_t = \mathcal{M}_j + \frac{[(\rho e)_t - \mathcal{E}] [v - 2ab_j]}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2\mathbf{a}\hat{\mathbf{b}}\cdot\vec{\mathbf{V}}} \quad (4.49)$$

Equation (4.47) is solved for density and equations (4.48) and (4.49) are solved for the components of momentum at subsonic exit boundary points to advance the solution in time. The value of $(\rho e)_t$ is determined during the iterative solution process.

Based on the perfect gas and no body forces assumptions discussed in Section II, the static pressure and the total energy are related in the following manner:

$$P = (\gamma-1)[\rho e - \frac{1}{2}\rho V^2] \quad (4.50)$$

This static pressure must be the specified pressure at the subsonic exit, P_{exit} .

The following iterative process is used to calculate the flow properties at the subsonic exit points. The process is initiated by guessing a value for the energy, ρe , for the new time level. For the first time step, the energy value is obtained from the initial-value surface. Subsequently, the initial guess is set at the energy value of the current time level.

The current time level value of energy and the value of energy guessed for the forward time level are used to numerically calculate a value for the temporal derivative of energy. This temporal derivative is used in equations (4.47) to (4.49) to calculate values for the density and the components of momentum. These values of density and momentum and the guessed forward time value of energy are used in equation (4.50) to calculate a value for pressure. If the calculated value of pressure is equal to the specified exit pressure, to within a small tolerance, the energy value guess was correct and the calculated values of ρ , ρu , ρv , and the guessed value of p_e are the property values at the forward time. If the calculated value of pressure does not agree with the specified exit pressure, a zero finding secant method is used to home in on the correct value of p_e . For the results presented in Section VII, a tolerance of 10^{-6} was used to evaluate convergence of the pressure difference normalized by the inflow total pressure.

IV.6.2 SUPERSONIC EXIT BOUNDARY POINT UNIT PROCESS

At an exit boundary point, when the component of the fluid velocity perpendicular to the exit boundary is greater than the local speed of sound, the fluid mechanics at the exit are influenced only by upstream conditions. In that case the interior point unit process is appropriate. Therefore, the MacCormack method is applied directly to the transformed governing equations in the strong conservation-law form, equation (2.33). The resulting finite difference equations are used to advance the solution in time.

IV.7 GRID CUT POINTS

All points on the grid cut lie in one of the regions discussed above. Specifically, the trailing edge of the blade, one exit point and several interior points make up the grid cut on a C-type grid. Therefore, no new equations need to be derived for the points on the grid cut. However, two factors complicate the solutions of all points on the grid cut.

1. Each physical grid point on the grid cut is represented by two noncoincident grid points on the computational grid.
2. In the computational space, where the calculations are carried out and where flow property values are stored, there are only grid points in the positive η direction from the grid cut. Therefore, the MacCormack method corrector calculation can not be carried out without special indexing.

These issues are resolved for the trailing edge point in Section IV.4.

The interior points are handled as described in Section IV.2 with a few additional steps included to resolve the problems listed above. For every $\xi = \text{constant}$ grid line intersecting the lower side of the grid cut, the $(\xi_{\max} + 1 - \xi) = \text{constant}$ grid line intersects the same physical location on the upper side of the grid cut. All grid cut calculations are computed at the computational grid points which lie on the lower side of the grid cut. Therefore, the property values at the $(\xi, 1)$ and $(\xi, 2)$ grid points are used to predict the η derivatives at the interior points which lie on the grid cut. Then, the predicted property values at the $(\xi, 1)$ and $(\xi_{\max} + 1 - \xi, 2)$ grid points

are used to correct the η derivatives at the interior points which lie on the grid cut.

After the property values have been predicted at $(\xi, 1)$, these predicted values are transferred to the $(\xi_{\max}+1-\xi, 1)$ predictor value storage locations for use in the $(\xi_{\max}+1-\xi, 2)$ corrector calculations. After the corrector calculations have been completed, the corrected property values are transferred to the $(\xi_{\max}+1-\xi, 1)$ corrector value storage locations.

The exit point which lies on the grid cut is handled as described in Section IV.6, but, with the same additional indexing steps just described for the grid cut interior points.

IV.8 PERIODIC BOUNDARY POINTS

All points on the periodic boundary lie in one of the regions discussed above. Specifically, one exit point and several interior points make up the periodic boundary on a C-type grid. Therefore, no new equations need to be derived for the points on the periodic boundary. However, three factors complicate the flow solutions at the points on the periodic boundary.

1. Since the flow through a cascade is assumed to be exactly periodic, the property values at each point on the periodic boundary on the lower side of the grid must be exactly the same values found at the corresponding grid point on the periodic boundary on the upper side of the grid.
2. In the computational space, where the calculations are carried out and where flow property values are stored, there are only grid points in the

negative η direction from the periodic boundary. Therefore, the MacCormack method predictor calculation can not be carried out without special indexing.

3. At the periodic boundary grid point adjacent to the inflow boundary, there are two grid points upstream of this point along $\eta = \text{constant}$ grid lines. Therefore, a decision must be made concerning how to calculate the ξ space derivatives at these leading periodic boundary grid points.

The periodic boundary interior points are handled as described in Section IV.2 with a few additional steps included to resolve the problems listed above. For every $\xi = \text{constant}$ grid line intersecting the the periodic boundary on the lower side of the grid, the $(\xi_{\text{max}}+1-\xi) = \text{constant}$ grid line is the corresponding grid line on the upper side of the grid. All periodic boundary calculations are computed at the computational grid points which lie on the periodic boundary on the lower side of the grid. Therefore, the property values at the (ξ, η_{max}) and $(\xi_{\text{max}}+1-\xi, \eta_{\text{max}}-1)$ grid points are used to predict the η derivatives at the periodic boundary points. Then, the predicted property values at the (ξ, η_{max}) and $(\xi, \eta_{\text{max}}-1)$ grid points are used to correct the η derivatives at the periodic boundary points. After the corrector calculations have been completed, the corrected property values are transferred to the $(\xi_{\text{max}}+1-\xi, \eta_{\text{max}})$ corrector value storage locations.

The special periodic boundary points which lie adjacent to the inflow boundary are handled as discussed above with one additional consideration. Both predictor and corrector ξ space derivatives are calculated entirely on the

$\eta = \eta_{\max} = \text{constant}$ line on the lower side of the C-grid. In other words, property values at (ξ, η_{\max}) and $(\xi+1, \eta_{\max})$ are used to calculate the predictor space derivatives at (ξ, η_{\max}) . Therefore, property values at $(\xi_{\max}-\xi, \eta_{\max})$ are not used to approximate property derivatives at this special point.

The exit point which lies on the periodic boundary is handled as described in Section IV.6, but, with the same additional indexing steps just described for the normal interior points which lie on the periodic boundary.

SECTION V

GRID GENERATION

Two of the major problems facing an analyst when constructing the numerical solution of partial differential equations are: the numerical implementation of the boundary conditions along the boundaries of the physical space, and the selection of the finite difference mesh to represent the continuous physical space. The boundaries of the physical space do not generally lie along coordinate lines formed by an equally spaced orthogonal grid system. When first-order accuracy is acceptable, boundary conditions can be implemented along arbitrary lines in the physical space and/or computations can be carried out on variable meshes. However, when higher-order accuracy is necessary, the above factors present serious difficulties. This has led to the extensive use of coordinate transformations to map the boundaries of physical space onto coordinate lines of a transformed space and to map nonuniform, nonorthogonal grids in physical space into uniform, orthogonal grids in transformed space. Once computational and physical grids have been generated, the numerical solution is implemented using the appropriate transformed governing equations.

In this section the elliptic partial differential equation grid generation method chosen for the present investigation is discussed. More details are provided in Appendix E. Information describing the transformation of the governing equations is presented in Section II and in Appendix A.

V.1 POISSON-TYPE CASCADE GRIDS

In this research, a Poisson-type elliptic partial differential equation grid generator is used to generate physical grids around cascade blades. Poisson's equation, applied to the generic variable $f(x,y)$, is presented in equation (5.1).

$$\nabla^2 f = f_{xx} + f_{yy} = F(x,y) \quad (5.1)$$

When multidimensional grids of this type are generated, one Poisson equation is used for each space dimension.

For a two-dimensional Poisson-type grid generator, the following Poisson equations must be solved numerically:

$$\nabla^2 \xi = \xi_{xx} + \xi_{yy} = P \quad (5.2)$$

$$\nabla^2 \eta = \eta_{xx} + \eta_{yy} = Q \quad (5.3)$$

where P and Q are the nonhomogeneous source terms. The freedom to specify these source terms in each Poisson equation provides the capability to control the relative point distribution in the interior of the grid. For example, coordinate surfaces can be clustered near a solid boundary or near an inflow boundary. This capability, as it has been applied in the present investigation, is discussed in more detail in Section V.2.

Figure 5-1 is an example of the C-type grids used in the present investigation. Additional examples are presented in Appendix E. The name is descriptive of the shape of each of the members of the family of grid lines that surrounds the blade.

Solving equations (5.2) and (5.3) numerically would be simple if the computations could be carried out on the physical grid. Unfortunately, due to unequal spacing in this grid, only first-order accuracy can be achieved using this approach. Therefore, the Poisson equations are transformed to the equally-spaced orthogonal computational domain and solved there. The transformed Poisson equations are:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -I^2(Px_{\xi} + Qx_{\eta}) \quad (5.4)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -I^2(Py_{\xi} + Qy_{\eta}) \quad (5.5)$$

where $I = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}$ is the Jacobian of the inverse transformation and

$$\alpha = x_{\eta}^2 + y_{\eta}^2$$

$$\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2$$

For the C-type grid, all boundary point locations are specified as Dirichlet boundary conditions as described in Appendix E. A modified successive over-relaxation (SOR) technique is used to solve equations (5.4) and (5.5), thereby locating the interior points. Like a normal SOR procedure, the iterative corrections at each point are over-relaxed. However, corrections to the values

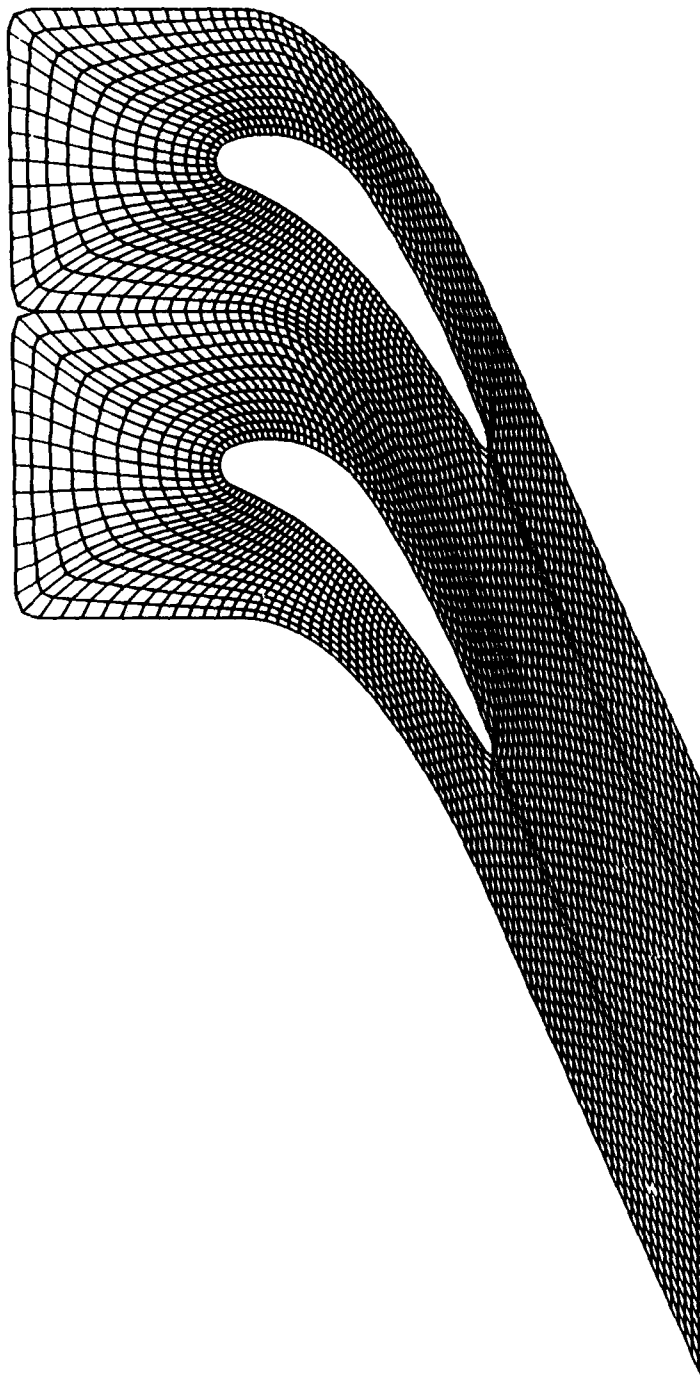


Figure 5-1. C-type grid for the AACE II cascade blade.

of the nonhomogeneous terms, P and Q , are under-relaxed each iteration. A description of these nonhomogeneous terms is presented next.

V.2 CONTROL OF GRID GEOMETRY NEAR BOUNDARIES

For the grids generated to support this investigation, the nonhomogeneous terms in the Poisson equations are used to control grid geometry based on the features of the grid at the $\eta = 1$ and $\eta = \eta_{\max}$ boundaries. Specifically, along the blade surface, grid cut, and inflow boundaries, the following two features are controlled:

1. Grid line spacings from points on the inner and outer boundaries to the corresponding points on the first grid line inside each of these boundaries are specified.
2. Intersection angles between these same boundaries and the $\xi = \text{constant}$ grid lines intersecting them are specified.

Similarly, along the periodic boundaries the following two features are controlled:

1. Grid line spacings from points on the inner and outer boundaries to the corresponding points on the first grid line inside each of these boundaries are specified.
2. Slope of the $\xi = \text{constant}$ grid lines in the physical plane where they cross the periodic boundaries is specified. The slope used is the tangent of the cascade stagger angle.

By dictating these specific grid features, grid point locations throughout the interior of the grid are controlled.

V.2.1 GRID CONTROL AT BLADE SURFACE, INFLOW, AND GRID CUT BOUNDARIES

Steger and Sorenson [1] have developed techniques to implement the controls described above. Some modifications to their method have been developed for this research effort.

Equations (5.4) and (5.5) are solved for P_b and Q_b , which are the P and Q values on the $\eta = 1$ and $\eta = \eta_{\max}$ boundaries.

$$P_b = J(y_\eta R_1 - x_\eta R_2) \Big|_{\eta_{\text{boundary}}} \quad (5.6)$$

$$Q_b = J(-y_\xi R_1 + x_\xi R_2) \Big|_{\eta_{\text{boundary}}} \quad (5.7)$$

where

$$R_1 = -J^2(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}) \Big|_{\eta_{\text{boundary}}}$$

$$R_2 = -J^2(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}) \Big|_{\eta_{\text{boundary}}}$$

Values of P_b and Q_b are calculated at each point on each of the $\eta = \text{constant}$ boundaries using equations (5.6) and (5.7). Since all points on both of these boundaries are specified, x_ξ , y_ξ , $x_{\xi\xi}$, and $y_{\xi\xi}$ are calculated from input data.

The key to Steger and Sorenson's method is their evaluation of x_η and y_η . They derive expressions for x_η and y_η by first considering the desired offset distance, ΔS , which is the distance in the physical space from the boundary

point, along a $\xi = \text{constant}$ grid line, to the corresponding point one grid line inside the boundary. Thus,

$$\Delta S = [(\Delta x)^2 + (\Delta y)^2]^{1/2} \Big|_{\xi = \text{constant}} \quad (5.8)$$

In the limit:

$$dS = [(dx)^2 + (dy)^2]^{1/2} \Big|_{\xi = \text{constant}} \quad (5.9)$$

From the partial differentiation chain rule,

$$dS = [(x_{\xi} d\xi + x_{\eta} d\eta)^2 + (y_{\xi} d\xi + y_{\eta} d\eta)^2]^{1/2} \Big|_{\xi = \text{constant}} \quad (5.10)$$

Since $dS/d\eta$ represents the distance between two points on the same $\xi = \text{constant}$ line, $d\xi = 0$. Therefore, equation (5.10) reduces to:

$$\frac{dS}{d\eta} = [(x_{\eta})^2 + (y_{\eta})^2]^{1/2} \Big|_{\xi = \text{constant}} \quad (5.11)$$

The second grid geometry control feature which Steger and Sorensen address is the grid line intersection angle at the periodic boundary. This control is implemented using the vector dot product:

$$\nabla \xi \cdot \nabla \eta = |\nabla \xi| |\nabla \eta| \cos \theta \quad (5.12)$$

where θ is the desired intersection angle ($\theta = 90^\circ$ yields orthogonal grid line intersections). Equation (5.12) is the definition of the dot product of the vector normal to the $\xi = \text{constant}$ grid line with the vector normal to the $\eta = \text{constant}$ grid line. Carrying out the vector arithmetic and incorporating the transformation metrics ($\xi_x = J y_{\eta}$, $\xi_y = -J x_{\eta}$, $\eta_x = -J y_{\xi}$, and $\eta_y = J x_{\xi}$) into equation (5.12) yields:

$$x_{\xi}x_{\eta} + y_{\xi}y_{\eta} = -[(x_{\eta}^2 + y_{\eta}^2)(x_{\xi}^2 + y_{\xi}^2)]^{1/2} \cos\theta \quad (5.13)$$

Solving equations (5.11) and (5.13) simultaneously to determine the necessary values of x_{η} and y_{η} required to meet these two conditions yields:

$$x_{\eta} = \frac{dS}{d\eta} \frac{(-x_{\xi} \cos\theta - y_{\xi} \sin\theta)}{(x_{\xi}^2 + y_{\xi}^2)^{1/2}} \quad (5.14)$$

$$y_{\eta} = \frac{dS}{d\eta} \frac{(-y_{\xi} \cos\theta + x_{\xi} \sin\theta)}{(x_{\xi}^2 + y_{\xi}^2)^{1/2}} \quad (5.15)$$

Both θ and $dS/d\eta$ are input discretely as functions of ξ on each of the $\eta = \text{constant}$ boundaries. Once the values of x_{η} and y_{η} on the boundaries are known, the cross derivatives $x_{\xi\eta}$ and $y_{\xi\eta}$ needed in equations (5.6) and (5.7) are calculated numerically by differencing x_{η} and y_{η} with respect to ξ . Therefore, the only remaining unknowns on the right-hand side of equations (5.6) and (5.7) are $x_{\eta\eta}$ and $y_{\eta\eta}$. The approach used by Steger and Sorenson is to numerically calculate $x_{\eta\eta}$ and $y_{\eta\eta}$ during each SOR iteration.

Once all the necessary derivatives have been determined, P_b and Q_b are calculated using equations (5.6) and (5.7). For the sake of stability, P_b and Q_b corrections are severely under-relaxed during each SOR iteration.

In order to smoothly propagate the P and Q effects throughout the grid, these quantities are exponentially extrapolated into the interior of the grid using equations (5.16) and (5.17).

$$P(\xi, \eta) = P(\xi, 1) e^{-a(\eta-1)/(\eta_{\max}-1)} + P(\xi, \eta_{\max}) e^{-b(\eta_{\max}-\eta)/(\eta_{\max}-1)} \quad (5.16)$$

$$Q(\xi, \eta) = Q(\xi, 1) e^{-c(\eta-1)/(\eta_{\max}-1)} + Q(\xi, \eta_{\max}) e^{-d(\eta_{\max}-\eta)/(\eta_{\max}-1)} \quad (5.17)$$

The terms $a = a(\xi)$, $b = b(\xi)$, $c = c(\xi)$, and $d = d(\xi)$ are positive decay rates used to control the exponential decay of the P and Q effects into the interior of the grid. Equations (5.16) and (5.17) differ slightly from those developed by Steger and Sorenson [1] in that the exponents have been normalized by $(\eta_{\max}-1)$. A zero finding secant method is used to home in on the appropriate decay rates at each value of ξ on each of the $\eta = \text{constant}$ boundaries. Since the theory behind this technique is developed in the limit of infinitely small space increments, but is applied to a finite size grid, the exact grid control requested will be approached, but not achieved exactly.

V.2.2 GRID CONTROL AT PERIODIC BOUNDARIES

A variation of Steger and Sorenson's technique has been developed during this investigation for the periodic boundaries of the cascade grid.

By forcing the $\xi = \text{constant}$ grid lines to cross the periodic boundary at the cascade stagger angle, γ , unnecessary grid line skewness is avoided. Since this $\xi = \text{constant}$ grid line slope control is more appropriate along the periodic boundary than $\xi = \text{constant}$ grid line intersection angle, the following equations replace equations (5.14) and (5.15):

$$x_{\eta} = \frac{dS}{d\eta} \sin(\gamma) \quad (5.18)$$

$$y_{\eta} = \frac{dS}{d\eta} \cos(\gamma) \quad (5.19)$$

SECTION VI

SOLUTION PROCEDURE

In the present investigation, the flowfield solution procedure consists of three major steps. First, grid points are placed around the boundary of the C-type grid. Second, the Poisson-type grid solver is used to locate all interior points. Third, the MacCormack flow solver is used to determine properties throughout the flowfield. The boundary point placement procedure and the grid solution procedure are described in Section V and in more detail in Appendix E. The flow solver solution procedure is described in this section.

The steady state flowfield solution is calculated as the asymptotic time dependent solution. Prior to the first time step, an initial property value surface is established. The calculation of initial property values is discussed in Section VI.1.

Because the grid is stationary, transformation Jacobian and metric values are calculated once, prior to the first time step, and used at every time step. As demonstrated in Appendix A, in order to avoid erroneous source terms, transformation metrics must be differenced in the same manner as property derivatives are differenced. Therefore, one-sided forward-differences are used to calculate metric values used in predictor calculations, and one-sided

backward-differences are used to calculate metric values used in corrector calculations. Centered-differences are used to calculate Jacobian values for both steps of the MacCormack calculations.

VI.1 INITIAL-VALUE SURFACE

Prior to beginning the time stepping procedure, property values are assigned to each grid point in the flowfield. The closer these initial values are to the final, steady state solution, the quicker the solution will converge to the steady state. Property values are first assigned on all the computational boundaries, in accordance with the appropriate boundary condition constraints. Values for the interior points are then interpolated from the boundary values.

The inflow total pressure and the exit static pressure are used to compute an approximate exit boundary Mach number, and in turn, the velocity magnitude at the exit boundary. The inflow total temperature, total pressure and the exit Mach number estimate are used to calculate the exit boundary density and total energy. The velocity components are specified by assuming the flow to be parallel to the grid cut at the exit boundary. Momentum components are the product of the respective velocity components and the density. All points on the exit boundary and on the grid cut, including the trailing edge, are set to these exit boundary property values.

The x-direction velocity component on each point on the periodic and inflow boundaries is assumed to be equal to the value of u calculated for the exit boundary. On the periodic boundary, v is then specified by assuming the

flow to be parallel to that boundary. On the inflow boundary v is set in accordance with the input inflow angle. Using the velocity magnitude and the inflow total properties, the Mach number is calculated at each point on these two boundaries. Then using the Mach number and the total properties, the density and energy are calculated at each point.

On the blade surface, the x-direction component of velocity is again assumed to be equal to the exit value except at the leading edge where a stagnation region is specified. The y-direction component of velocity is then assigned so that the velocity is everywhere parallel to the blade surface. The density and total energy are then determined as described above.

Once all relevant property values have been calculated at all boundary points, interior point values are determined by interpolating, between boundaries, along $\xi = \text{constant}$ grid lines.

VI.2 TIME STEPPING PROCEDURE

The flow solver time stepping procedure consists of smoothing the property values at the current time level, calculating the appropriate time step at every grid point, predicting and correcting solutions to the inviscid flow equations which are appropriate for each point, and testing the solution for convergence.

As described in Section IV.4, both the predictor and the corrector calculations are included in the iterative process at the trailing edge point. Therefore, to advance the solution in time, the following sequence is established. All points on the grid, except the trailing edge point are

predicted. Then the trailing edge point is iteratively predicted and corrected. Finally, all remaining points are corrected. This process is repeated, advancing the solution in time, until convergence has been achieved at all grid points, to within a specified tolerance.

VI.2.1 TIME STEP CALCULATION

The magnitude of the time step used to calculate properties is regulated by the Courant Friedrichs Lewy (CFL) stability criterion [24]. Enforcing the CFL restriction insures that the domain of dependence of the partial differential equations (the Mach cone) falls within the domain of dependence of the finite difference equations (the convex hull). The convex hull for the MacCormack method is outlined with bold lines in Figure 6-1.

In the present investigation, a conservative approximation is used to estimate the CFL limit which is appropriate for each grid point. At each point, the distance between the point and the nearest point on the convex hull is calculated. Since the grid is stationary, the minimum distance corresponding to each grid point is calculated once, stored, and then used each time the time step limits are recalculated. The velocity vector is assumed to point away from the closest point on the convex hull. Based on this assumption, the desired time step limit is:

$$\Delta t_{\text{CFL}} = \frac{d}{|V| + a} \quad (6.1)$$

where d is the shortest distance to the convex hull boundary, $|V|$ is the

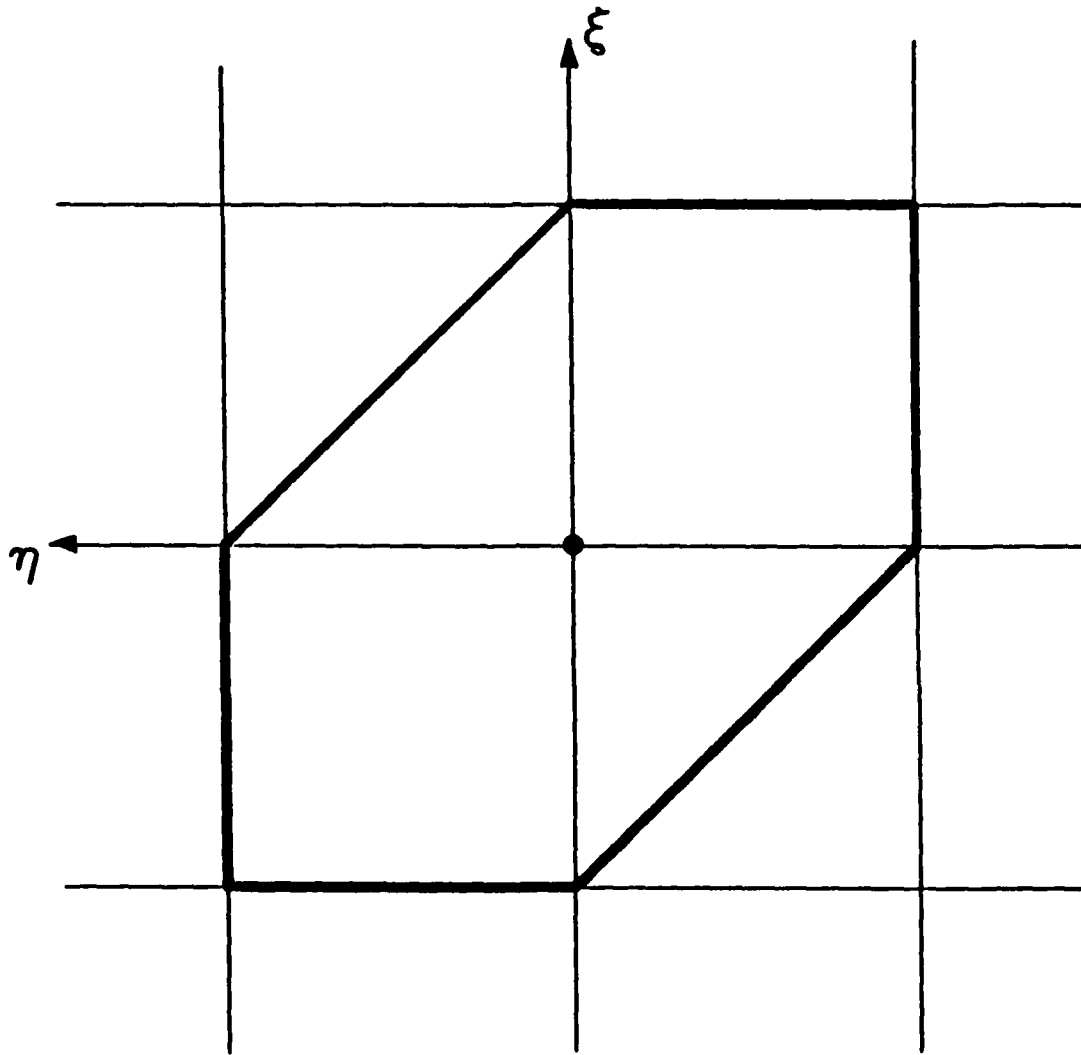


Figure 6-1. The two-dimensional MacCormack method convex hull.

velocity magnitude, and a is the local speed of sound.

The local CFL time step limit can be calculated and used at each grid point. This local time step procedure produces a stable and rapidly converging solution. However, the transient is not accurate with respect to time since different time steps are applied at each grid point. In order to achieve a stable and time accurate solution, the smallest time step calculated for any point on the grid (global time step) must be used at every point on the grid to advance the solution in time. Both options are available in the codes developed in the present investigation.

In order to provide the flexibility to limit the time step further, time step multiplying factors are used. Two time step multiplying factors are input. Generally the initial time step multiplying factor, with a value around 0.7, is used to promote stability for the potentially rough initial-value surface. After a specified number of time steps, a time step multiplying factor of around 1.0 is substituted and used to advance the solution to convergence.

Because the properties in the flowfield are changing, the CFL limit at each grid point changes in time. Since a conservative CFL estimate is being used, computational effort is reduced by updating the time step limits after several (usually 5) time steps, rather than after every time step.

VI.2.2 EXPLICIT SMOOTHING

A numerical solution to the unsteady Euler equations can be limited by stability considerations. In other words, if the numerical solution to a set of

unsteady partial differential equations diverges rather than predicting a bounded solution to the equations, no useful information is derived.

As discussed in the previous section, a stability analysis for a particular numerical method applied to a particular set of partial differential equations produces a time step limitation which, if exceeded, will lead to divergence. Unfortunately, common stability analysis techniques only predict the stability criterion applicable to a simplified linear model equation in the absence of the boundaries of the computational domain. Therefore, even when the calculated stability restrictions are enforced, instabilities can still occur. In particular, when numerically solving the Euler equations, nonlinearities in the equation formulations, various boundary condition implementations, and flow solution phenomena such as shock waves and stagnation points, often cause numerical instabilities. Explicit artificial dissipation, or smoothing, is usually applied to overcome these instabilities. An extensive discussion of artificial dissipation is provided in Appendix F. The specific smoothing techniques used in the present investigation are summarized here.

The current research effort employs a conservation variable form of the Kentzer method at the flowfield boundaries and the MacCormack method throughout the flowfield. This combination requires some explicit dissipation to avoid divergence. Specifically, the blade surface boundary appears to generate instabilities. However, when a small amount of smoothing is applied over the majority of the grid, excellent results are achieved.

The general formula used to smooth each of the four dependent conservation variable properties ρ , ρu , ρv , and ρe , is:

$$f_{\text{smoothed}} = f + \Delta t c_2 \left[sf\xi_2 f_{\xi\xi} + sf\eta_2 f_{\eta\eta} \right] \quad (6.2)$$

where f is one of the solution properties, Δt is the time step, c_2 is a property independent variable coefficient, and $sf\xi_2$ and $sf\eta_2$ are grid dependent scaling factors.

All the smoothing terms in equation (6.2) are multiplied by the time step. This causes the smoothing to be scaled at the same magnitude as the space derivatives in the Euler equations. In addition, if local time steps are being used rather than global time steps, multiplication by the time step will cause more explicit artificial dissipation to be added where larger time steps may be allowing instabilities to grow faster.

The property independent coefficient, c_2 , sets the magnitude of the smoothing for the entire flowfield. In order to keep a rough initial-value surface from diverging and yet avoid excessively smearing the final solution, this coefficient is linearly decreased in time. The upper and lower limits of c_2 are input along with a multiplying factor which is set to approximately 0.999. At each time step, the value of c_2 is multiplied by this factor until the lower limit of c_2 is reached. At all subsequent time steps c_2 is maintained at the lower limit.

The grid dependent scaling factors, $sf\xi_2$ and $sf\eta_2$, are used to efficiently correct unwanted, grid dependent scaling of the dissipation derivatives, $f_{\xi\xi}$ and

$f_{\eta\eta}$. Without these factors, the dissipation derivatives have less of an effect in areas of the physical grid where grid points are densely packed than in areas where grid points are sparse. It is desirable to eliminate this effect without the added computational effort of computing transformed physical derivatives at every time step. Therefore, a procedure has been developed to rescale property derivatives which are taken with respect to computational directions. As explained in Appendix F, the following equations for $sf\xi_2$ and $sf\eta_2$ provide the desired effect.

$$sf\xi_2 = \frac{4}{(x_{i+1,j} - x_{i-1,j})^2 + (y_{i+1,j} - y_{i-1,j})^2} \quad (6.3)$$

$$sf\eta_2 = \frac{4}{(x_{i,j+1} - x_{i,j-1})^2 + (y_{i,j+1} - y_{i,j-1})^2} \quad (6.4)$$

The cascade grids used for the present investigation do not change with time. Also, $sf\xi_2$ and $sf\eta_2$ are only functions of grid point locations. Therefore, $sf\xi_2$ and $sf\eta_2$ are calculated once and stored, prior to the first time step. These stored values are then used at every time step.

At the blade surface and trailing edge boundaries, smoothing is only applied tangent to the boundaries. At these boundaries the momentum magnitude is smoothed and then split into x and y components to insure tangency.

No smoothing is applied at the inflow or at the exit boundaries.

Interior points on the periodic boundaries and grid cut boundaries are smoothed like other interior points, but with special indexing required due to

the locations of these boundaries on the computational grid.

VI.2.3 CONVERGENCE CHECK

With unchanging boundary conditions, sufficient time stepping will lead to asymptotic values for all properties at all grid points. This set of property values is the steady state solution for the flowfield.

In the present investigation, experimentation with the code reveals that the total energy is the slowest of the four solution variables to converge. Therefore, changes in the total energy are monitored to determine the level of convergence.

At the end of each time step, the change in total energy at each grid point is calculated, normalized by the total energy at that grid point, and compared to an input tolerance value to determine if the solution has converged sufficiently. For all results described in Section VII, solutions were converged to a tolerance of 10^{-5} .

Care is taken to avoid the effects of smoothing on the convergence check. Prior to smoothing the solution at any time level, the newly calculated energy values are stored and are compared to the energy values stored at the previous time step.

SECTION VII

RESULTS

The final step in the development of the Euler solver in the present investigation is the evaluation of the capability of the code. This evaluation includes testing the code over the range of relevant cascade environments. To this end, eight numerical case studies have been completed and evaluated. One case was chosen as the baseline test case. All other cases differ from the baseline case by varying one or more parameters. The results of those case studies are discussed in this section.

VII.1 NUMERICAL TEST CASES

Table 8.1 summarizes the case studies used to evaluate the flow solver. The following three parameters were varied:

1. Blade geometry.
2. Mach number regime.
3. Grid density.

The Mach number regime was varied by varying the ratio of inflow total pressure to exit static pressure.

Table 7-1. Cascade numerical test cases.

Test Case	Cascade	Numerical Method	Grid Dimensions $\xi \times \eta$	Mach Number Regime	Time Step Mode (Factor)	Inflow Total Press. ¹ Total Temp. ² Flow Angle ³	Exit Static Press. ¹
1 ⁴	AACE II	MacCormack	165 x 13	Transonic	Local (1.0)	96904.0 300.0 0.0	55981.0
2	AACE II	M O C ⁵	165 x 13	Transonic	Local (1.0)	96904.0 300.0 0.0	55981.0
3	AACE II	MacCormack	83 x 7	Transonic	Local (1.0)	96904.0 300.0 0.0	55981.0
4	AACE II	M O C	83 x 7	Transonic	Local (1.0)	96904.0 300.0 0.0	55981.0
5	AACE II	MacCormack	165 x 13	Subsonic	Local (1.0)	96904.0 300.0 0.0	66327.0
6	AACE II	MacCormack	165 x 13	High Transonic	Global (0.65)	96904.0 300.0 0.0	37308.0
7	GMA 400	MacCormack	195 x 11	Subsonic	Local (1.0)	96904.0 300.0 42.4	64392.0
8	GMA 400	MacCormack	195 x 11	High Subsonic	Local (1.0)	96904.0 300.0 42.6	53976.0

1 Pascals

2 Kelvin

3 degrees

4 baseline case

5 numerical method of characteristics

Two different cascades were evaluated in the present investigation. The AACE II [6] and the GMA 400 [7] cascades were chosen for several reasons. First, both of these cascades are representative of modern turbine blade geometries. In addition, the cascades differ significantly from each other. Finally, high quality experimental blade surface pressure data are available for both of these cascades.

In order to meet the needs for maximum power extraction, minimal shock losses, and light weight component designs, turbines are usually designed to operate with maximum Mach numbers only slightly in excess of Mach one. Therefore, to verify the ability of this code to evaluate the relevant range of turbine Mach numbers, test cases were chosen with maximum Mach numbers varying from approximately 0.8 to 1.35.

The MacCormack method is second-order accurate in time and space. Therefore, the density of the grid used to discretize the flowfield has a great effect on the accuracy of the results of the numerical studies. In order to evaluate the dependence of the flowfield solutions on grid point density, two different grid densities were used. For most test cases, grids with 2145 grid points (165x13 or 195x11) were used. For two of the test cases involving the AACE II cascade, a grid with 581 (83x7) grid points was used. Grid points were placed with similar relative point density distributions on both AACE II grids. The coarse grid represents a decrease in grid density of approximately half in each of the computational directions. This decrease should result in a four fold increase in numerical error.

The three grids discussed above are illustrated in Figures 7-1 to 7-3. Figure 7-1 is the high density (165 x 13) AACE II grid. Figure 7-2 is the low density (83 x 7) AACE II grid. Figure 7-3 is the high density (195 x 11) GMA 400 grid. The dimensions of the two high density grids differ because the solidity (chord length over blade spacing) of the GMA 400 cascade (1.83) is greater than the solidity of the AACE II cascade (1.30).

The predictions of all of the test cases were compared against experimental results. The experimental results consisted of blade surface static pressure measurements for each of the cascades. Because the experiments were all run with air at approximately standard sea level pressure and temperature, the numerical studies were run under the same conditions.

The numerical method of characteristics is recognized as an excellent numerical method because it so accurately matches the fluid physics it models. An inviscid numerical method of characteristics cascade flow solver has been developed as part of this research effort. The characteristics code is described in Appendix C. This code is used in the present investigation as another verification of the primary, MacCormack code. Therefore, the results of two of the primary code test cases mentioned above are compared to the results of the numerical method of characteristics code.

VII.1.1 TEST CASE 1

Case 1 is chosen as the baseline test case for the following two reasons. First, in addition to experimental results and results from the numerical method of characteristics code developed for the present investigation, the

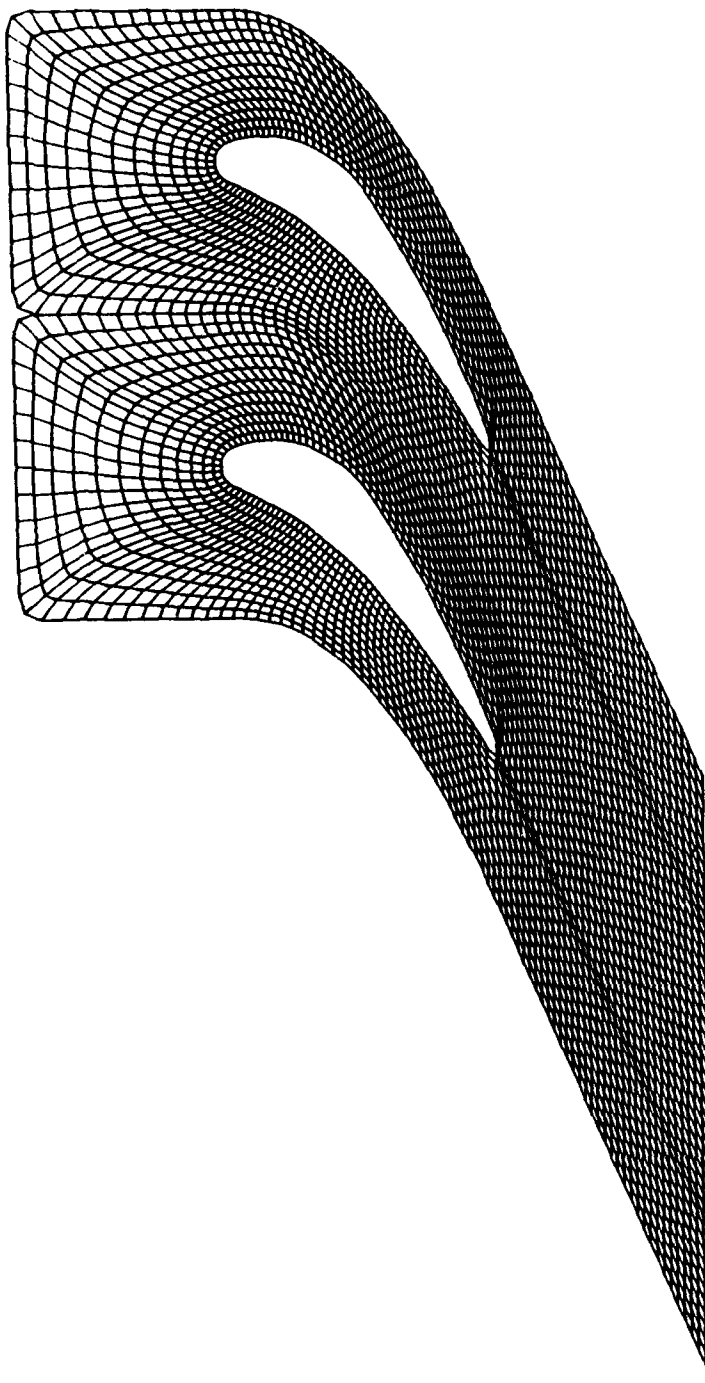


Figure 7-1. High density AACE II cascade grid. (165 x 13).

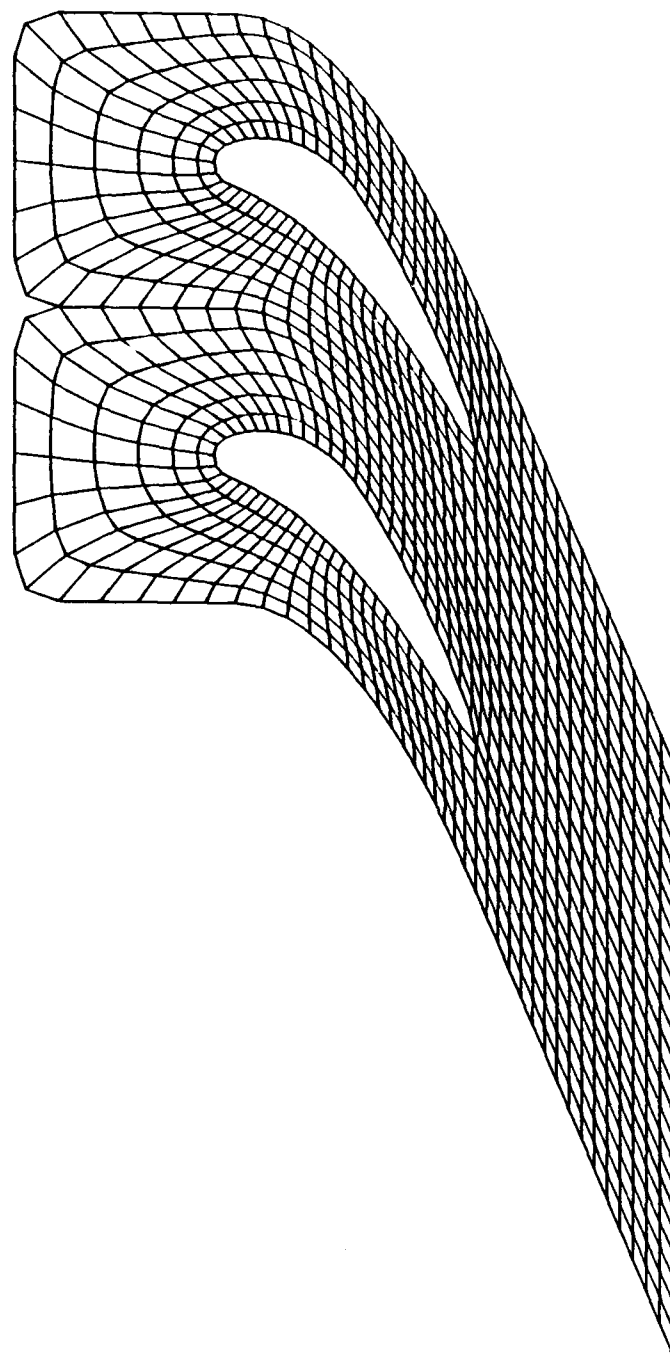


Figure 7-2. Low density AACE II cascade grid. (83 x 7).

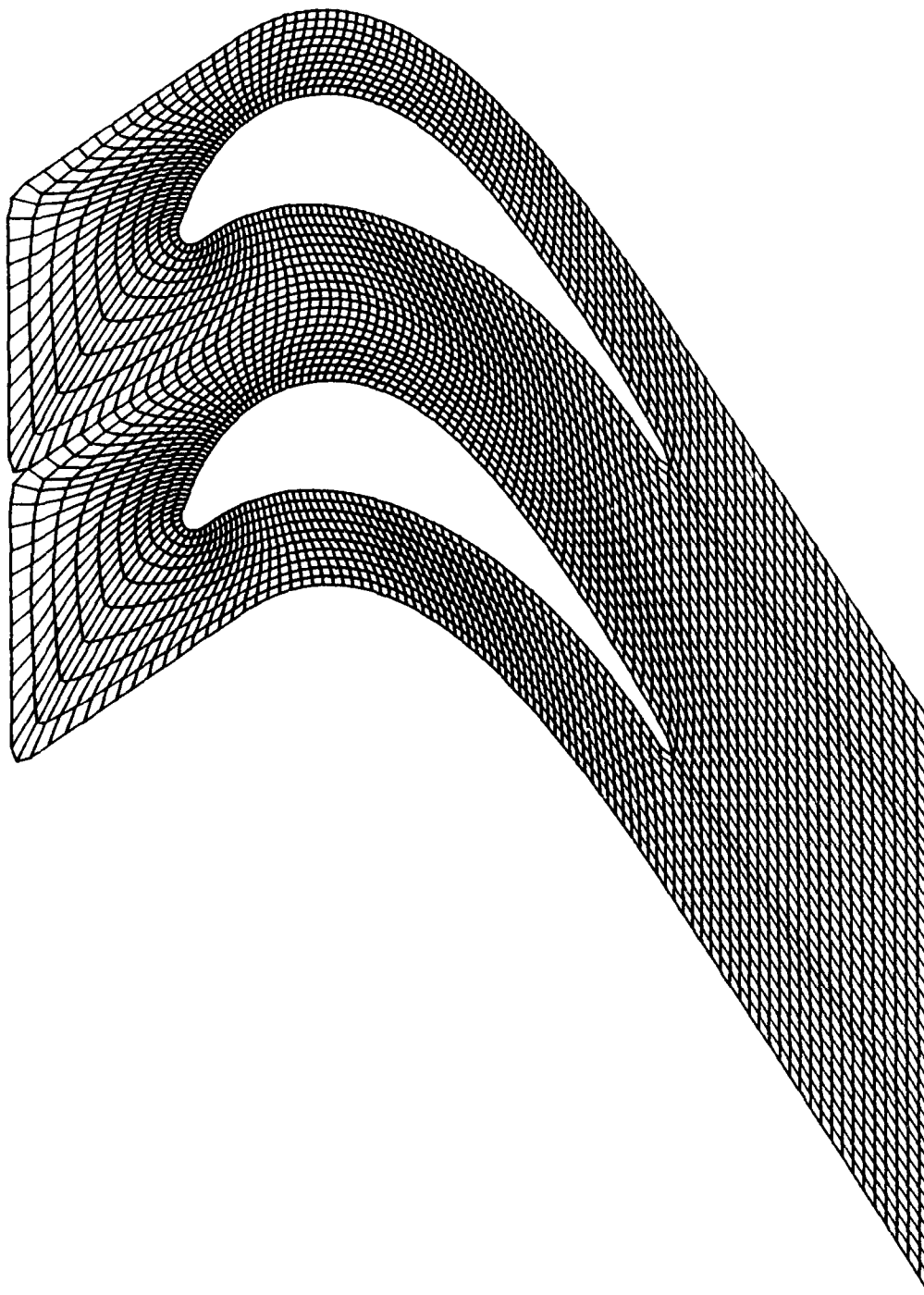


Figure 7-3. High density GMA 400 cascade grid. (195 x 11).

AACE II blade has been studied by Delaney [14] and Kemry [15]. Second, the pressure ratio across the blade row predicts an average exit Mach number of 0.9211. This suggests that flow slightly in excess of Mach one should be expected in the flow passage near the suction side of the blade. This is representative of turbine blade row design conditions.

In addition to being representative of turbine design conditions, the range of Mach numbers present in the baseline case flowfield provides an excellent test of the numerical scheme. For this case, a large region between the cascade blades contains fluid flowing at Mach numbers between 0.9 and 1.1. Property gradients are large in flowfield regions where this range of Mach numbers exist. When flowfields are solved numerically, high property gradients tend to cause correspondingly high numerical errors. Therefore, the accuracy with which fluid properties are calculated in these regions of the flowfield demonstrates the capability of a numerical method.

Figure 7-4 presents the blade surface pressure for the Case 1. As illustrated, the numerically calculated blade surface static pressure distribution agrees quite well with the experimental results. Over the first seventy percent of the pressure side of the cascade blade, the pressure values match almost exactly. From the seventy percent point to the trailing edge region the calculated pressures slightly exceed the measured values.

Both the experimental and numerical results show two low pressure (high speed) regions on the suction side of the blade. Between the midchord of the suction side and the trailing edge region of the pressure side, the flow is

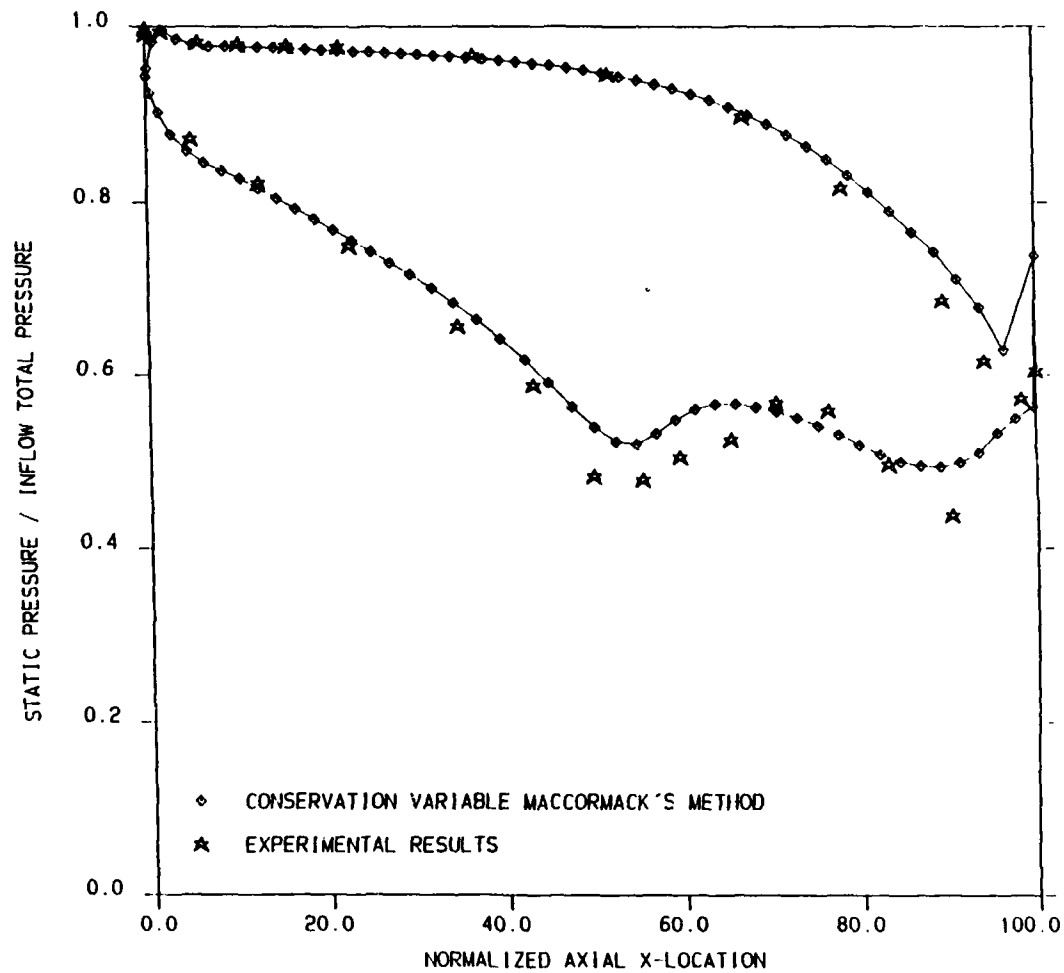


Figure 7-4. Test Case 1. Blade surface normalized static pressure.
AACE II cascade, baseline case.

accelerated because of the flow passage restriction. Just upstream of the trailing edge the flow is again accelerated. Figures 7-5 and 7-6 illustrate flowfield static pressure and Mach number contours, respectively. The contours in each of these figures reveal these high speed, low pressure regions. Figure 7-6 reveals a small supersonic pocket at the second location. These two low pressure regions are evident, to some extent, in all test cases considered for the AACE II cascade. The calculated results predict slightly less pressure variation across the suction side of the blade than do the measured data.

The blade trailing edge is the only location where the calculated pressure differed drastically from the measured pressure. Accurate calculations at the trailing edge are virtually impossible, especially with an inviscid code. A review of the literature reveals that numerical analysts usually try to keep the trailing edge calculations from diverging and from adversely affecting the solution upstream of that region. See for example, the results of Delaney [16], Srivastava et al. [17], and Kwon [18].

One factor used to calculate turbine blade row work is the circumferential component of the force exerted by the fluid on the blade. This force can be calculated by integrating the circumferential component of the pressure difference between the two sides of the blade, over the axial component of the chord. Based on this method of evaluation, the pressure discrepancies generated at the trailing edge, by the present effort or any of the other efforts just cited, are completely acceptable. Further discussion of the trailing edge is presented in Section VII.3.

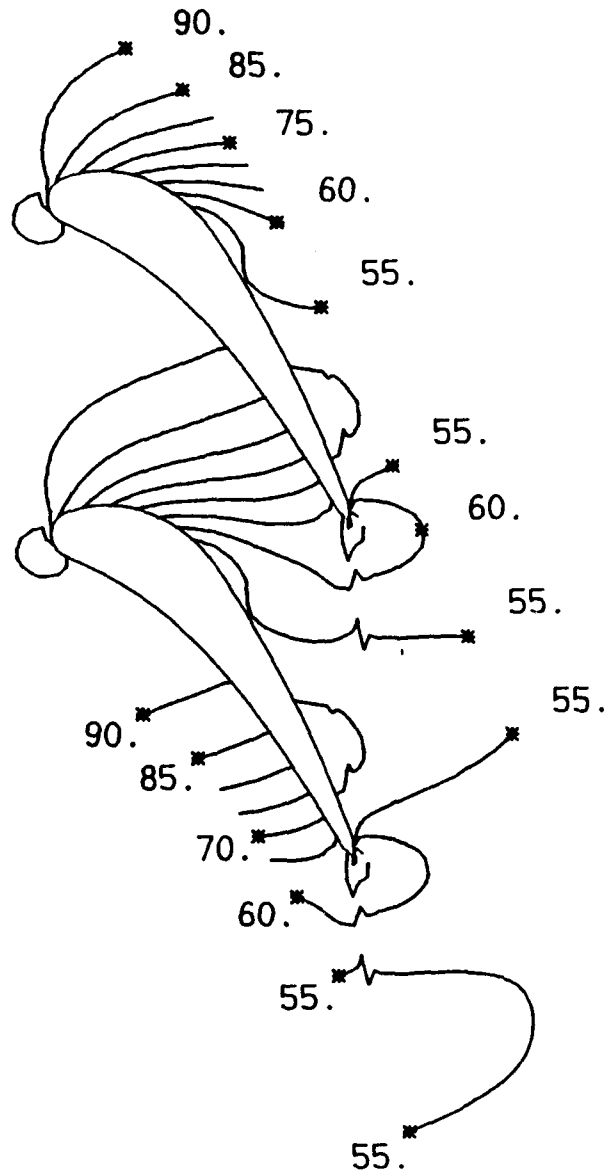


Figure 7-5. Test Case 1. Static pressure contours.

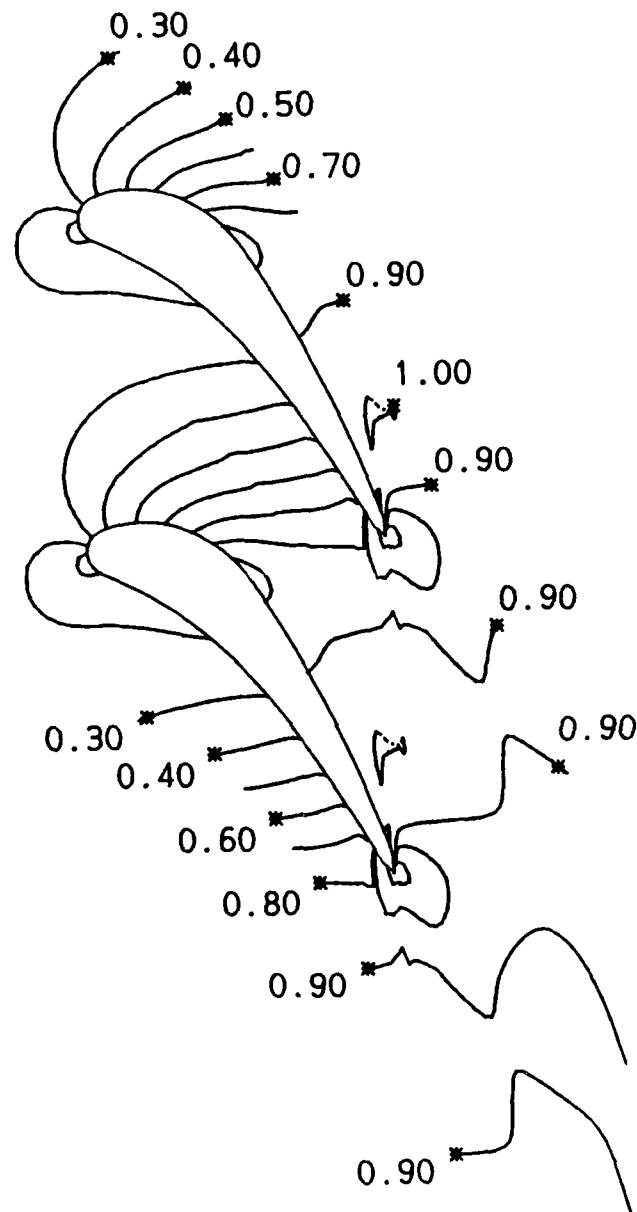


Figure 7-6. Test Case 1. Mach number contours.

VII.1.2 TEST CASE 2

The flow conditions of Case 2 are identical to those of Case 1. However, in Case 2 the numerical method of characteristics code, rather than the MacCormack code, has been used to predict the flow properties. The results are very similar to the results of Case 1.

As illustrated in Figure 7-7, the characteristics code predicts nearly exactly the same pressure distribution on the pressure side of the blade as the primary code. Similarly, the trailing edge point pressure prediction is close to that predicted by the primary code.

As compared to the experimental data, the characteristics code slightly under-predicts the pressure on the suction side of the blade, just aft of the leading edge. This discrepancy is also produced by the primary code, as shown in Figure 7-4, but at less than half the magnitude as by the characteristics code.

The pressure variations on the aft half of the blade are resolved more accurately by the characteristics code than by the primary code. Both codes show the greatest discrepancy between the computed surface pressures and measured surface pressures on the aft half of the suction side of the blades. Both under predict the pressure variations in that region. Specifically, both predict that the static pressure values in that region are slightly higher than the measured values. However, the characteristics code predicts the two low pressure regions on the suction side of the blade more accurately than the primary code.

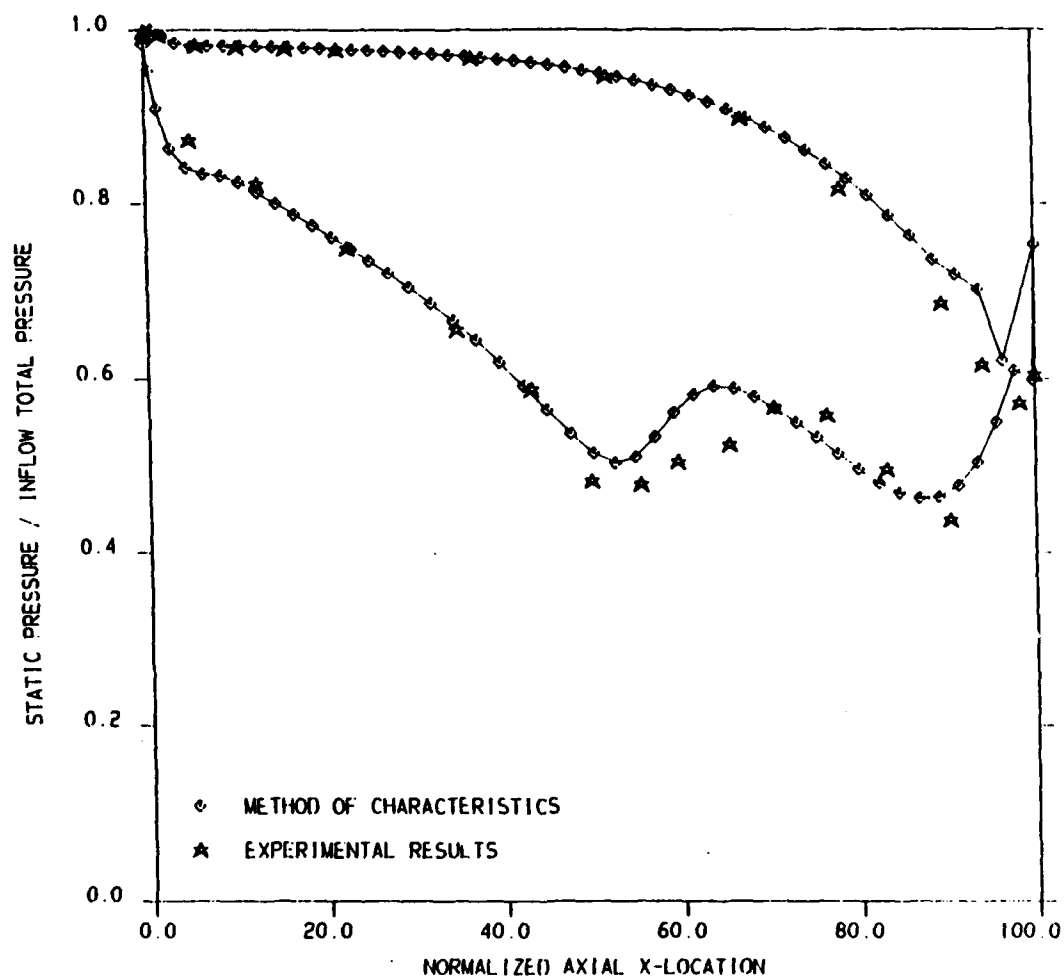


Figure 7-7. Test Case 2. Blade surface normalized static pressure.
AACE II cascade, method of characteristics.

Figures 7-8 and 7-9 illustrate static pressure and Mach number contours, respectively, for the flow passage. The pressure contours shown in Figure 7-8 reveal that the characteristics code predicts slightly lower pressures on the second half of the suction side of the blade as compared to the MacCormack code predictions. Figure 7-9 shows a larger supersonic bubble computed by the characteristics code in the high speed region just upstream of the trailing edge.

VII.1.3 TEST CASE 3

The flow conditions and the numerical method of Case 3 are identical to those of Case 1. However, in order to evaluate the accuracy of the method, the flow is solved on the coarse grid shown in Figure 7-2. In some regions of the flowfield, the results are very similar to the results of the baseline case. In other regions, significant resolution is lost on the coarse grid.

Figure 7-10 presents the blade surface pressure for Case 3. As this figure shows, the suction side pressure distribution predicted on the coarse grid is almost identical to that predicted on the fine grid. Some resolution very near the trailing edge is lost on the coarse grid. However, the static pressure predicted immediately at the trailing edge is closer to the measured value. This anomaly is addressed in Section VII.3.

The property value resolution at the leading edge is reduced on the coarse grid. Figure 7-10 reveals that the maximum static pressure predicted on the blade surface near the stagnation point at the leading edge is only 98 percent of the inflow total pressure.

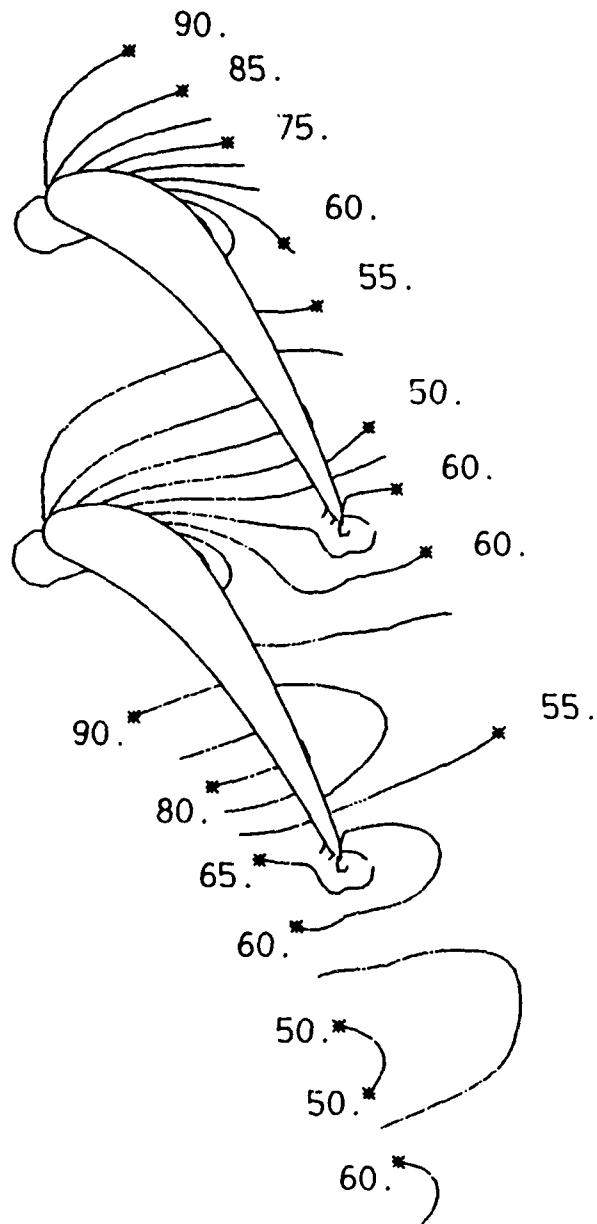


Figure 7-8. Test Case 2. Static pressure contours.

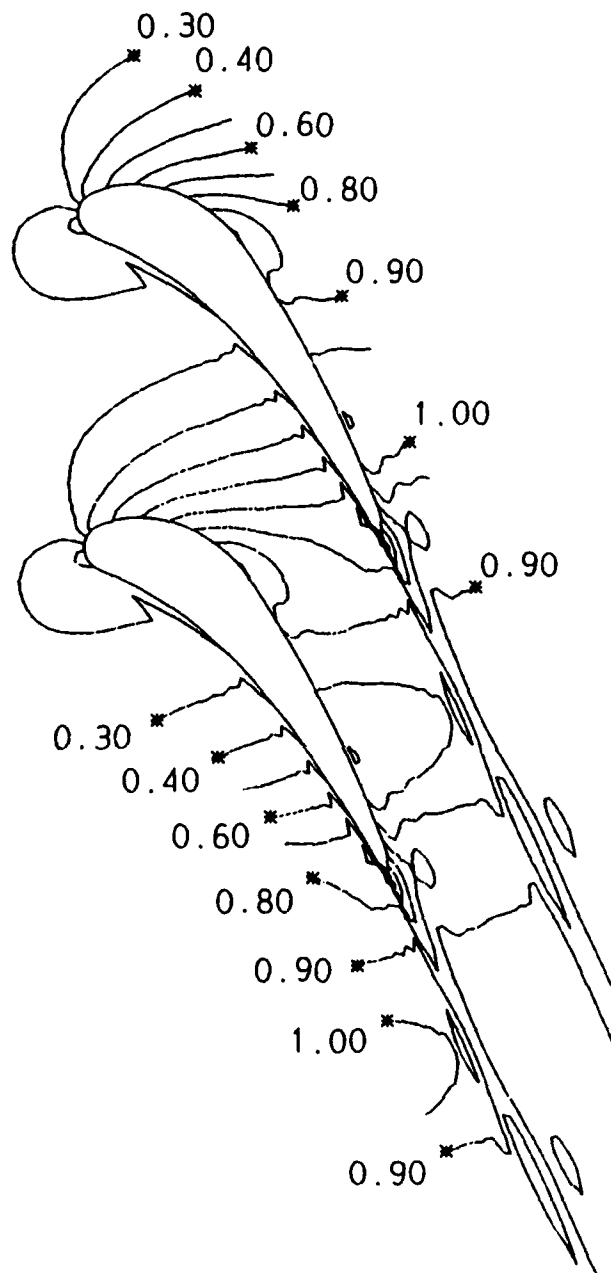


Figure 7-9. Test Case 2. Mach number contours.

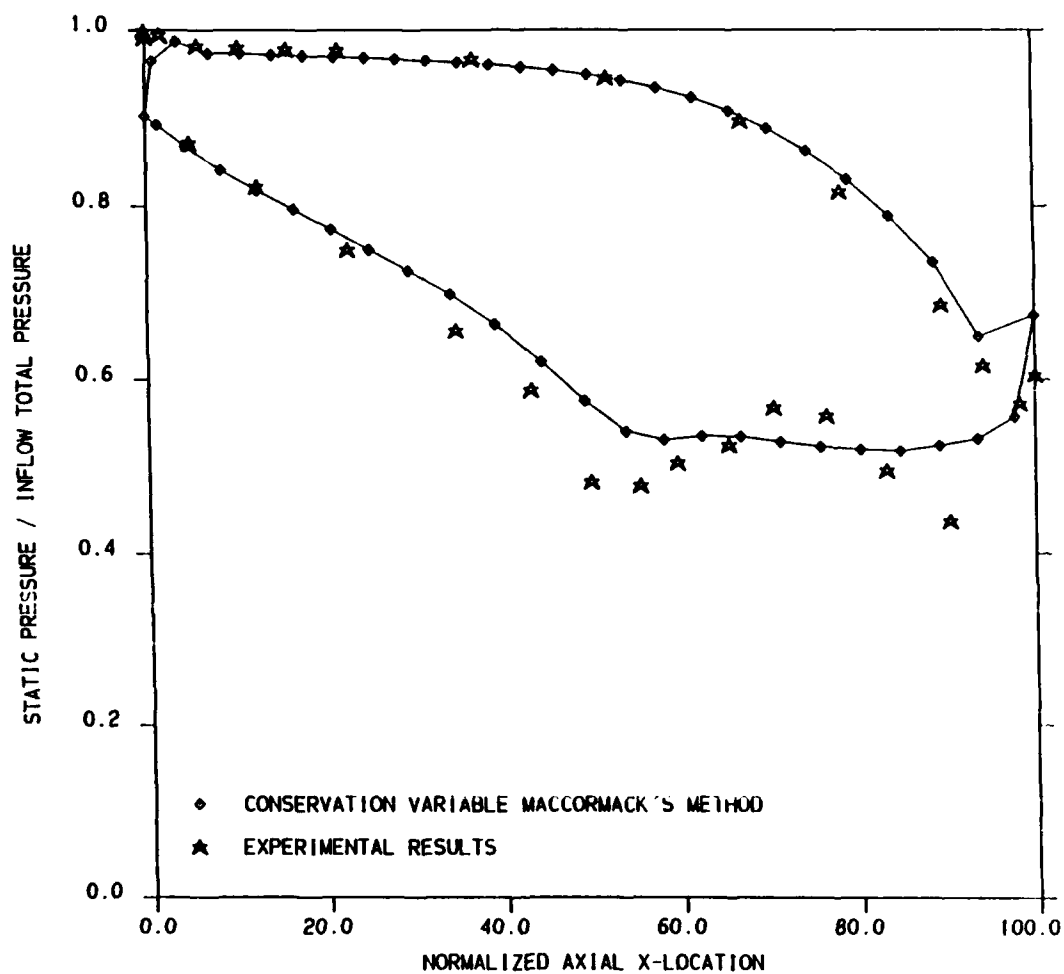


Figure 7-10. Test Case 3. Blade surface normalized static pressure.
AACE II cascade, coarse grid.

The calculated pressure distribution on the aft half of the suction side of the blade differs significantly from that predicted on the dense grid. The two distinct low pressure regions predicted by both codes on the dense grids and measured experimentally, are smeared together on the coarse grid. The code predicts a nearly constant pressure level over this part of the blade. This predicted constant pressure level is approximately equal to the actual average pressure for the region. However, the chordwise variation of pressure is lost.

Figures 7-11 and 7-12 illustrate static pressure and Mach number contours, respectively, for the flow passage. Figure 7-11 reveals the same loss of resolution discussed above. The pressure contours shown in this figure are very similar to those shown in Figure 7-5 except on the aft half of the suction side of the blade. The coarse grid solution shows just one large region of pressure between 50 kPa and 55 kPa. Similarly, the Mach number contours in Figures 7-6 and 7-12 are very similar except in the region of the flowfield near the aft half of the suction side of the blade.

Increased grid density also results in increased computational costs. Therefore, the improved resolution supplied by higher density grids must be considered in the light of added computational expenses. In addition, successively denser grids will provide diminishing improvements in flowfield resolution. Certainly, grid resolution is sufficient when significant increases in grid density produce relatively insignificant improvements in flowfield resolution. Because of the large increases in solution accuracy near the aft suction side of the blade, and to a lesser extent, near the leading edge stagnation point, it is obvious that the coarse grid is inadequate. Since the

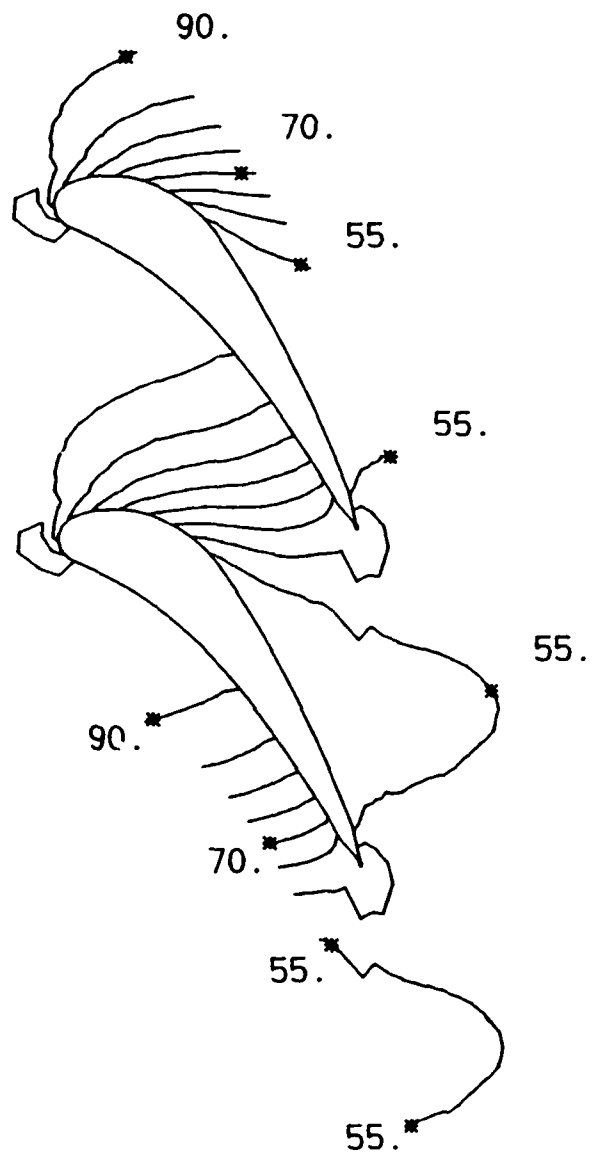


Figure 7-11. Test Case 3. Static pressure contours.

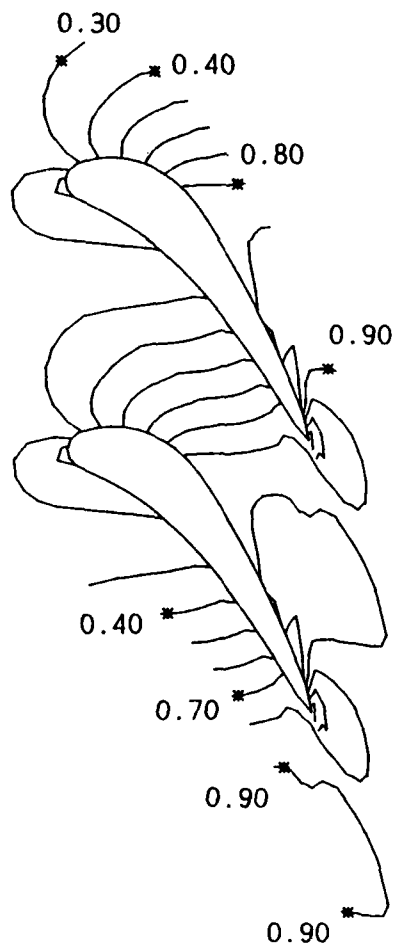


Figure 7-12. Test Case 3. Mach number contours.

denser grid resolves all the major flow features, it is adequate.

In addition to higher computational expenses, increasing the density past that of the dense grid shown in Figure 7-1 is constrained by considerations discussed in Section VII.3.

VII.1.4 TEST CASE 4

The flow conditions of Case 4 are identical to those of Cases 1, 2, and 3. Like Case 2, Case 4 compares the results of the primary code to those of the numerical method of characteristics. As before, the two numerical methods produce similar results, with some tradeoffs, but with slightly better performance by the characteristics code in the regions of the flowfield which have proven difficult to resolve accurately. Figures 7-13 to 7-15 illustrate the results obtained by the numerical method of characteristics solved on the low density grid.

Cases 1 through 4 all predict very nearly the same pressure distribution on the pressure side of the blade. On the other hand, the characteristics code resolves the leading edge stagnation point as well on the coarse grid as on the dense grid.

Unlike the primary code operating on the coarse grid, the characteristics code does distinguish the two low pressure regions on the suction side of the blade. However, the resolution is not as good as is produced by either of the numerical methods used in conjunction with the dense grid.

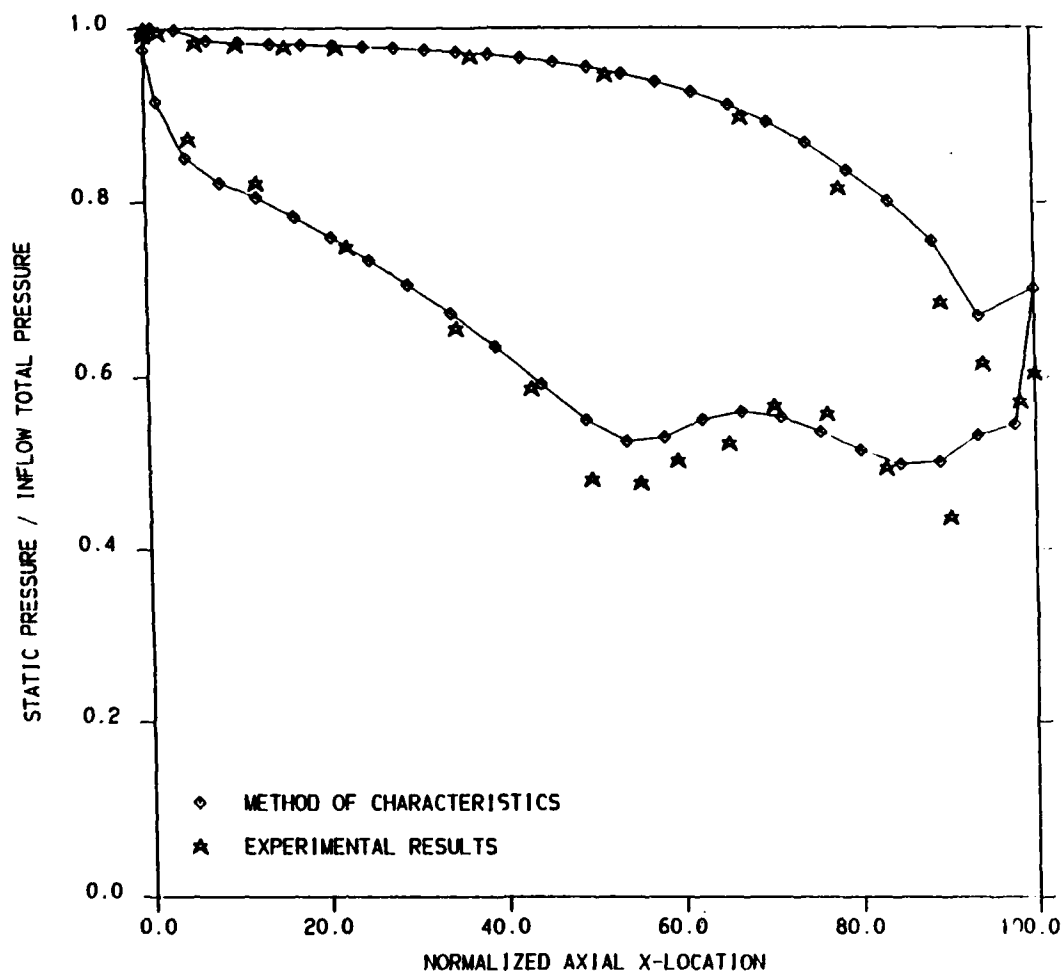


Figure 7-13. Test Case 4. Blade surface normalized static pressure.
AACE II cascade, method of characteristics, coarse grid.

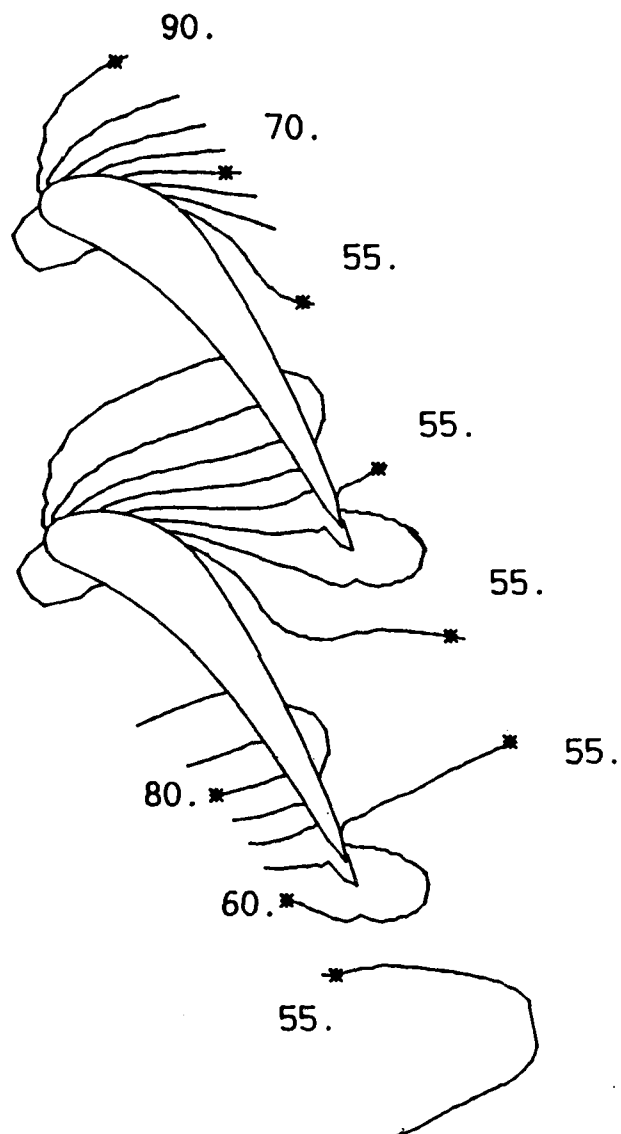


Figure 7-14. Test Case 4. Static pressure contours.

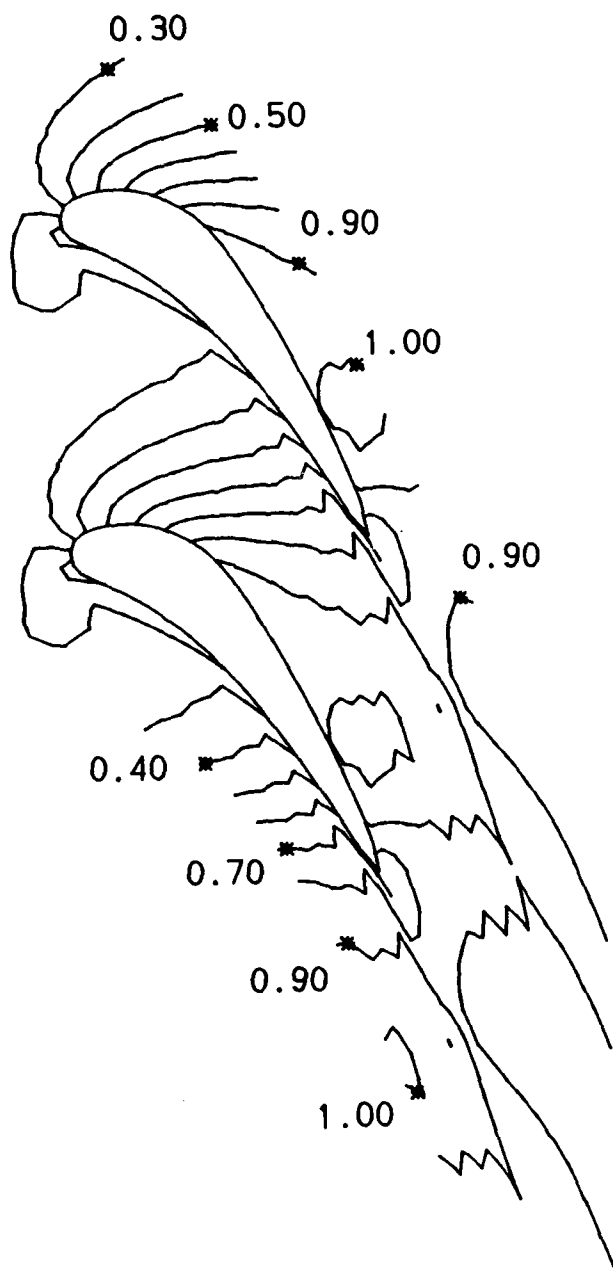


Figure 7-15. Test Case 4. Mach number contours.

VII.1.5 TEST CASE 5

Test Case 5 is the first case with flow conditions which differ from those of the baseline case, test Case 1. In order to test the code in a totally subsonic flowfield, the MacCormack code was run at a pressure ratio across the blade row which corresponds to an average theoretical exit Mach number of 0.76. As with Cases 1 and 2, this case is solved on the dense AACE II cascade grid shown in Figure 7-1.

Figure 7-16 presents the blade surface pressure for Case 5. As shown in this figure, the experimental results corresponding to this case differ from those of the baseline case in a few respects. The most noticeable difference is the shallowness of the low pressure region on the suction side of the blade, just upstream of the trailing edge. In the baseline case, this region contains the lowest pressure on the blade. In this case, the pressure is nearly constant over the last 35 percent of the blade.

The measured pressure distribution over the remainder of the blade, including the high speed region near the center of the suction side of the blade, very closely resembles the pressure distribution over the blade in the baseline case. As should be expected with a higher downstream pressure, the blade surface pressures measured for Case 5 are slightly higher than those measured for the baseline case.

Figure 7-16 demonstrates good agreement between the calculated and the measured blade surface pressures. The MacCormack code predicts pressures which are nearly exactly the blade surface pressures except in the low pressure region in the center of the suction side of the blade and on the aft thirty percent of the pressure side of the blade. In these regions, the calculated pressure is as much as six percent higher than the measured pressure.

The pressure and Mach number contours for Case 5 are presented in Figures 7-17 and 7-18. The maximum Mach number shown in Figure 7-18 is 0.80. This confirms that the entire flowfield is subsonic.

VII.1.6 TEST CASE 6

In contrast to Case 5, Case 6 tests the capability of the MacCormack code to calculate the flow through the AACE II cascade in the high transonic Mach number regime. For Case 6, the pressure ratio across the blade row corresponds to a theoretical average exit Mach number of 1.25. Again, the dense AACE II cascade grid, shown in Figure 7-1, is used to discretize the flowfield.

Figures 7-19 to 7-21 illustrate the results obtained for Case 6. As shown in these Figures, the experimental results corresponding to this case differ significantly from those of the previous cases, especially over the second half of the blade. As Figure 19 illustrates, the shape of the pressure distribution on the pressure side of the blade is similar to the previous cases, but the pressures are lower. On the suction side of the blade, the pressure drops well below any measured in the previous cases. Careful inspection of the measured pressures

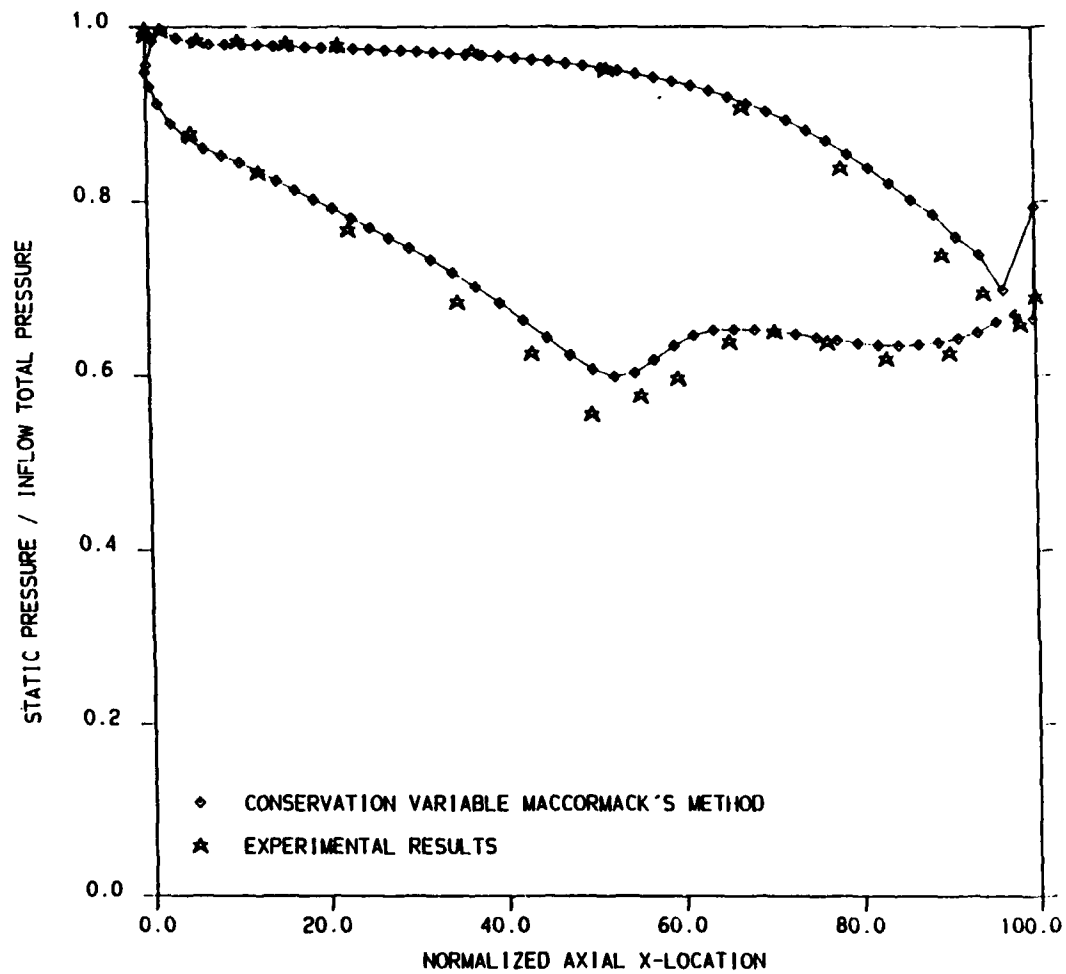


Figure 7-16. Test Case 5. Blade surface normalized static pressure. AACE II cascade, subsonic flowfield.

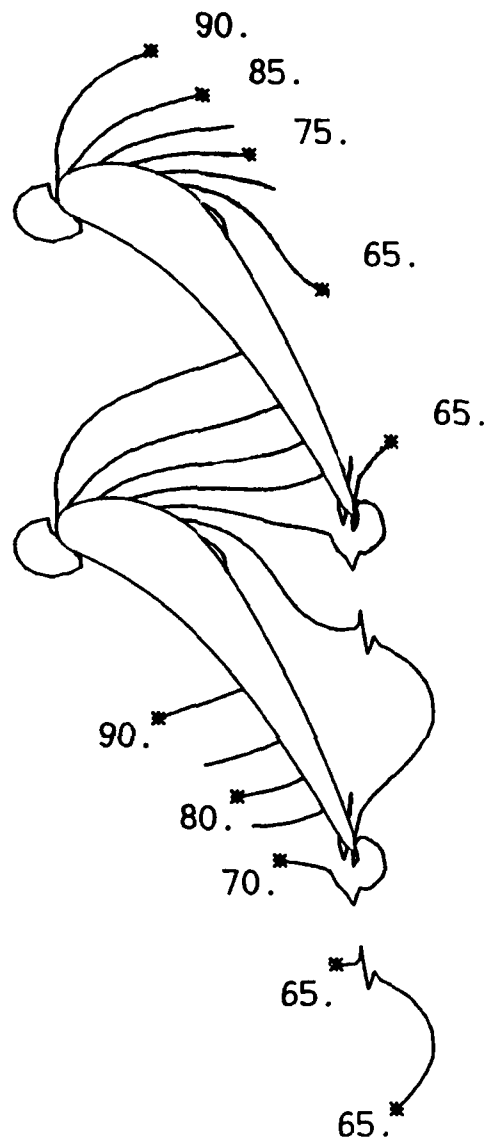


Figure 7-17. Test Case 5. Static pressure contours.

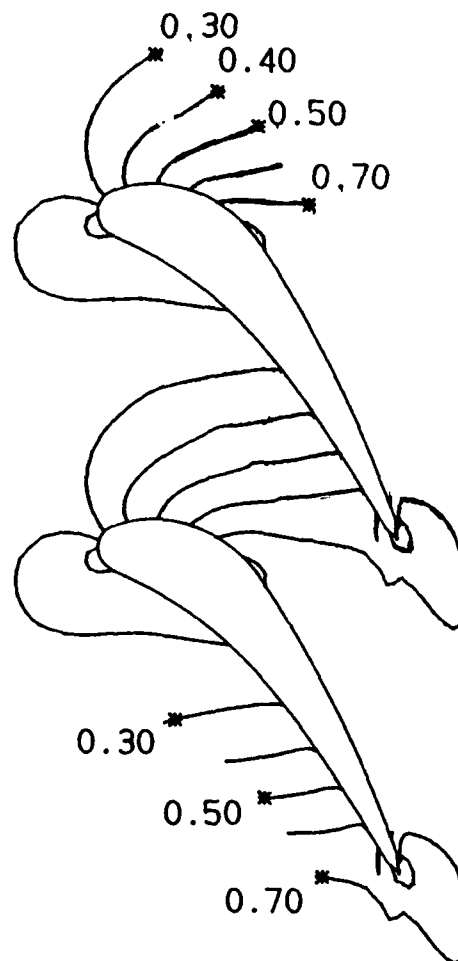


Figure 7-18. Test Case 5. Mach number contours.

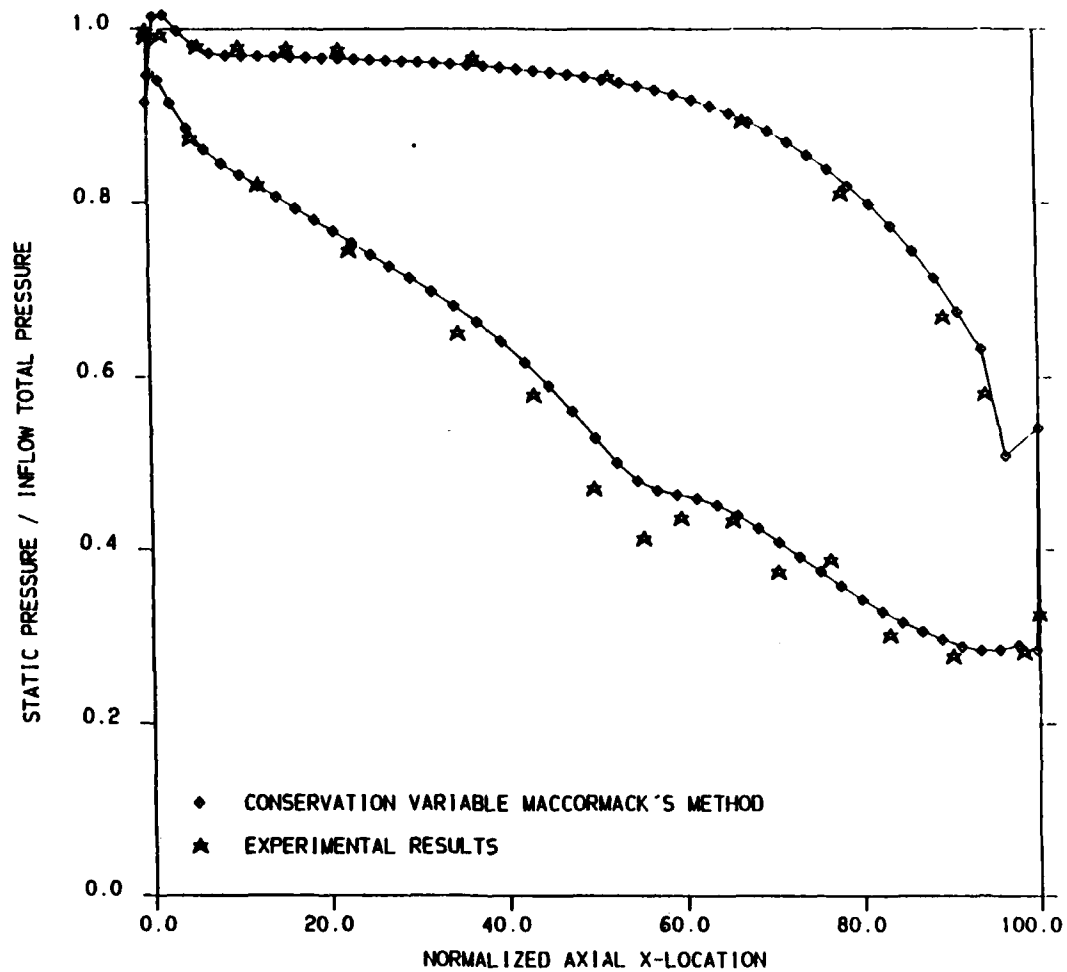


Figure 7-19. Test Case 6. Blade surface normalized static pressure.
AACE II cascade, high transonic flowfield.

on the suction side of the blade reveals three low pressure regions rather than two. This may represent an error in the measured values.

The calculated blade surface pressures shown in Figure 7-19 agree quite well with the measured values. Agreement on the pressure side of the blade is nearly exact. Along the suction side the calculated values follow the measured values, but are generally from zero to four percent above the measured pressure values. At the center of the first low pressure region, the calculated pressure is six percent higher than the experimental value.

At the blade leading edge the property gradients in Case 6 are slightly higher than in the previous cases. This becomes evident when Figures 7-20 and 7-21 are compared with previous pressure contour and Mach number contour plots. Apparently, these higher gradients increase the numerical instabilities on the leading edge of the blade. The total pressure overshoot and the low pressure spike, which are both shown near the upper left corner of Figure 7-19, are evidence of this unstable tendency.

Increasing the density of the grid near the leading edge is one means of increasing the stability of the solution in that region. However, in order to maintain consistency with the other test cases, this idea was not used. Another means of increasing the stability of the solution is to increase the magnitude of the explicit smoothing coefficients described in Section VI.2.2. Unfortunately, large increases in the smoothing coefficients can cause the resolution of the property variations in the final solution to be smeared excessively. A third means of increasing stability is to decrease the time step.

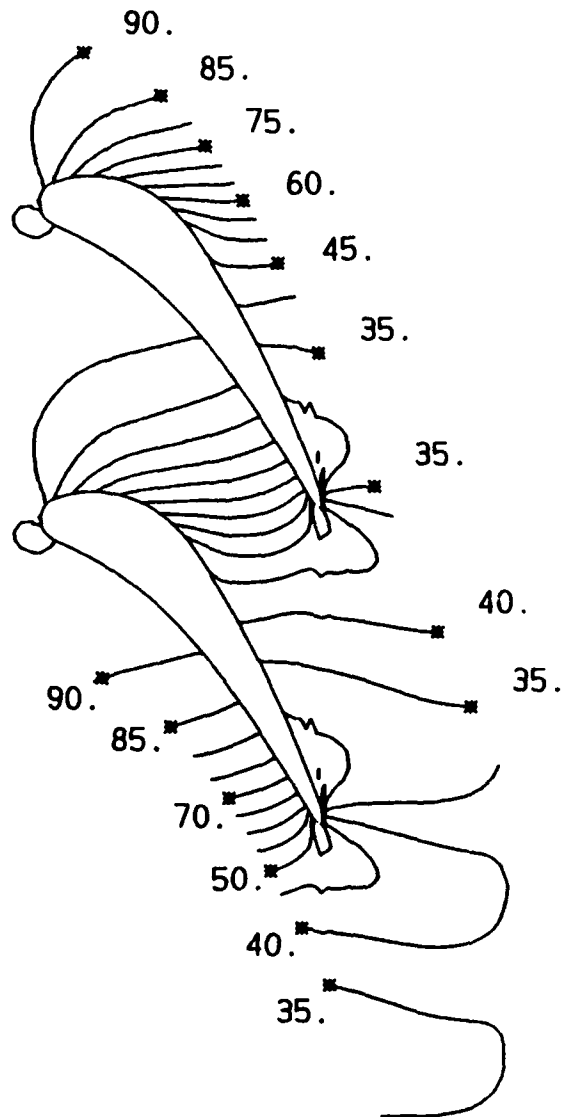


Figure 7-20. Test Case 6. Static pressure contours.

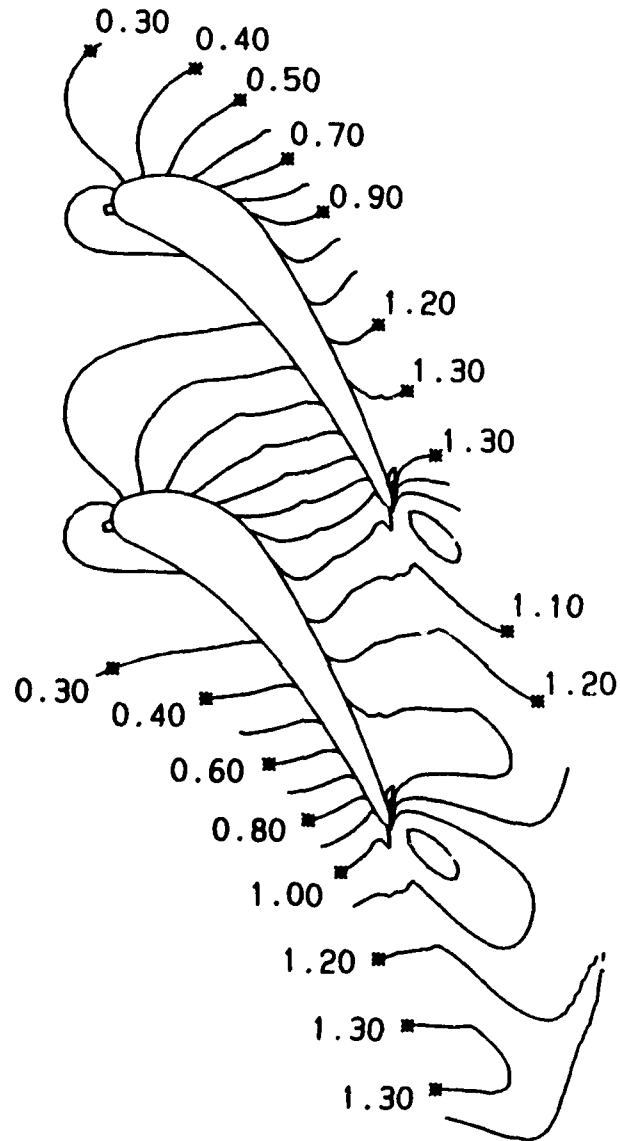


Figure 7-21. Test Case 6. Mach number contours.

Figure 7-12. Test Case 3. Mach number contours.

This effectively increases the implicit smoothing of the solution. For the present case, stability is achieved by slightly increasing the smoothing coefficients and by using a global time step limit instead of a local time step limit. In addition, the calculated time step is multiplied by a factor of 0.65.

If this high transonic regime was characteristic of turbine blade row flowfields, an extensive modification of the implemented smoothing technique might be necessary. Specifically, property dependent coefficients could be used to target instabilities at the leading edge. However, these coefficients would at least double the computations required to smooth the flowfield. Therefore, since this case is representative of the maximum Mach numbers found in turbine blade rows, the present smoothing scheme is retained.

Figure 7-21 confirms that the Mach number at the exit boundary is approximately 1.2. On the suction side of the blade, Mach numbers in excess of 1.3 are computed. Unlike the previous cases, the entire flowfield aft of the choke point is supersonic.

VII.1.7 TEST CASE 7

To demonstrate the capability of the MacCormack code to solve a different cascade flowfield, a distinctly different cascade geometry was chosen. The GMA 400 cascade, illustrated in Figure 7-3, was designed and tested by the Allison Gas Turbine Division of the General Motors corporation [7]. The flowfield through this cascade was calculated at two pressure ratios. Case 7 involves a flowfield with subsonic Mach numbers similar to that of Case 5.

The flow leaving a jet engine combustor is usually not swirling. Therefore, the zero leading edge camber angle of the AACE II cascade is representative of the first blade row in a turbine. The GMA 400 cascade is representative of a downstream blade row. The leading edge camber angle of this blade is 49.4 degrees. This allows the blades to operate efficiently in a flowfield which has a circumferential velocity component which is approximately equal to or slightly larger than the axial component. As Table 7.1 shows, the experimental data were collected with inflow angles of 42.4 and 42.6 degrees for Case 7 and Case 8 respectively. This corresponds to approximately minus 7 degrees of incidence. As Smith [7] notes, under these conditions viscous effects cause the flow to separate on the pressure side of these blades. Since the present investigation neglects viscous effects, some numerical errors should be expected on the pressure side of the blade.

Measured and calculated blade surface static pressures are presented in Figure 7-22. Agreement between the two sets of values are excellent over most of the blade. On the pressure side of the blade, where viscous separation occurs in the actual flow, the calculated pressure values are as much as three percent below the measured pressure values.

Figures 7-23 and 7-24 illustrate static pressure and Mach number contours, respectively, for the GMA 400 flow passage. As Figures 7-22, 7-23, and 7-24 show, the property gradients over the first 30 percent of the suction side of the blade are very high. This appears to be the cause of the higher than expected calculated pressure values on that portion of the blade surface. Figure 7-24 confirms that the entire flowfield is subsonic.

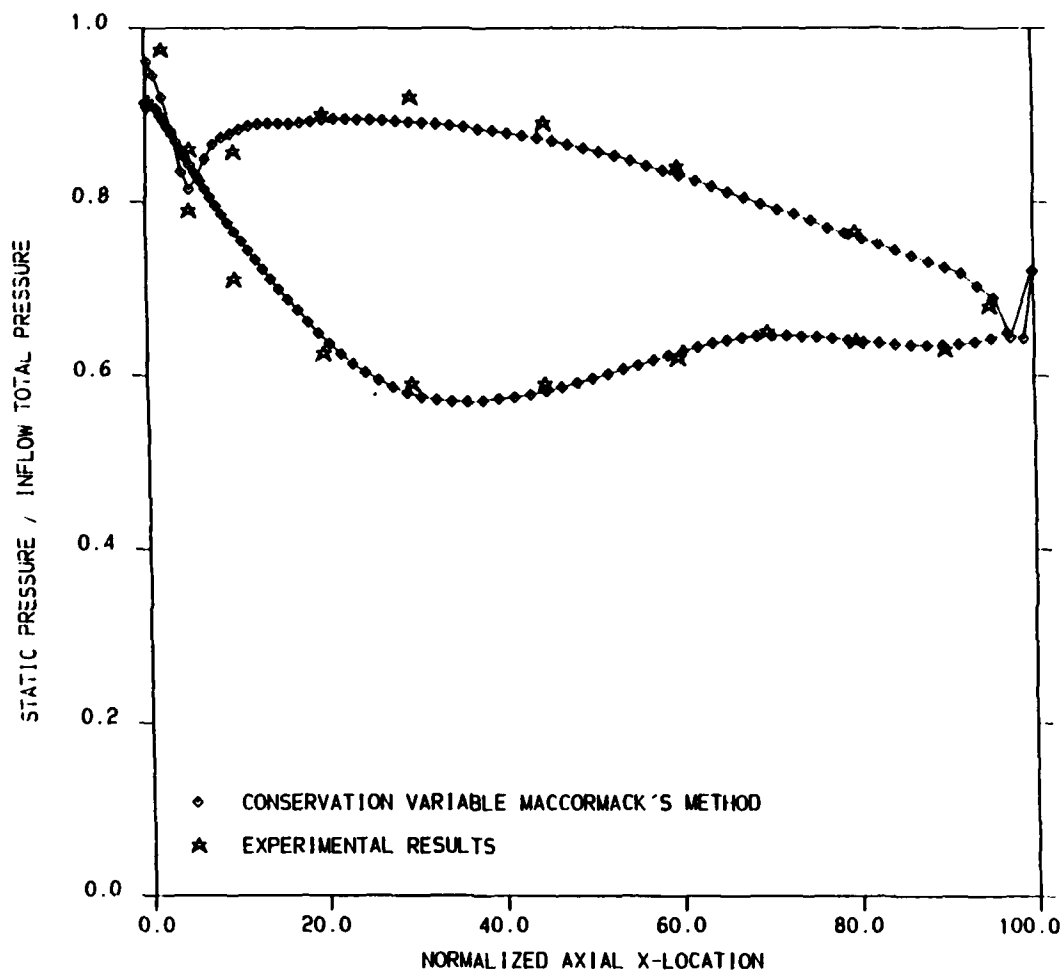


Figure 7-22. Test Case 7. Blade surface normalized static pressure.
GMA 400 cascade, subsonic flowfield.

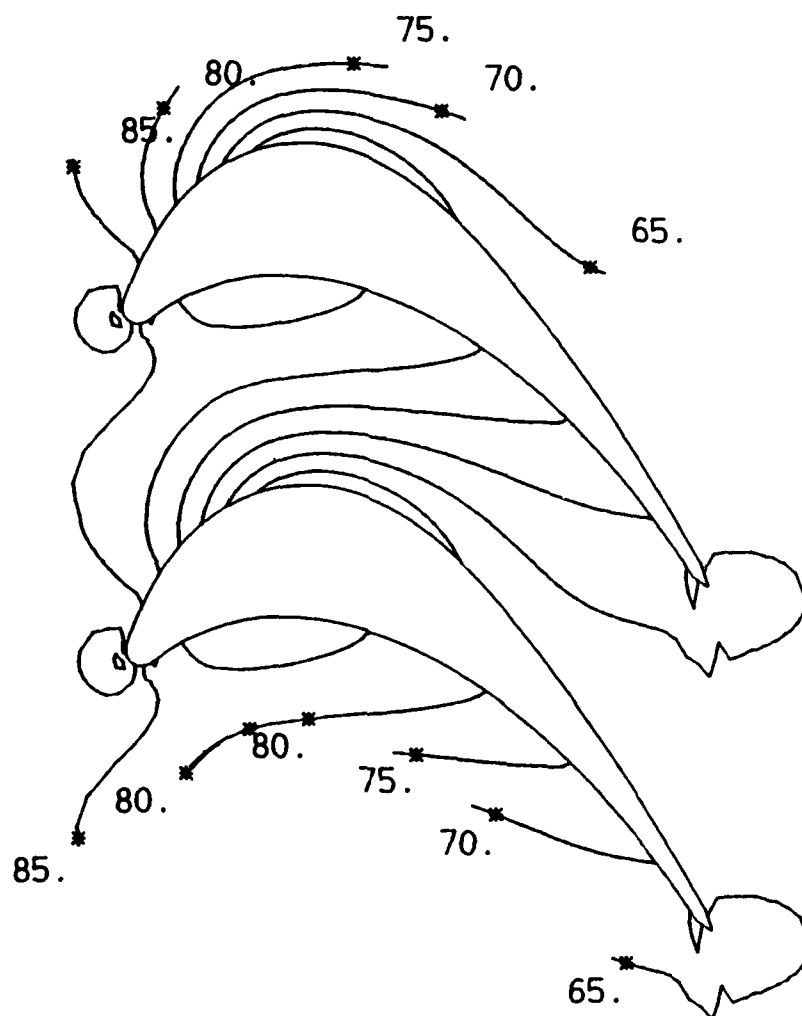


Figure 7-23. Test Case 7. Static pressure contours.

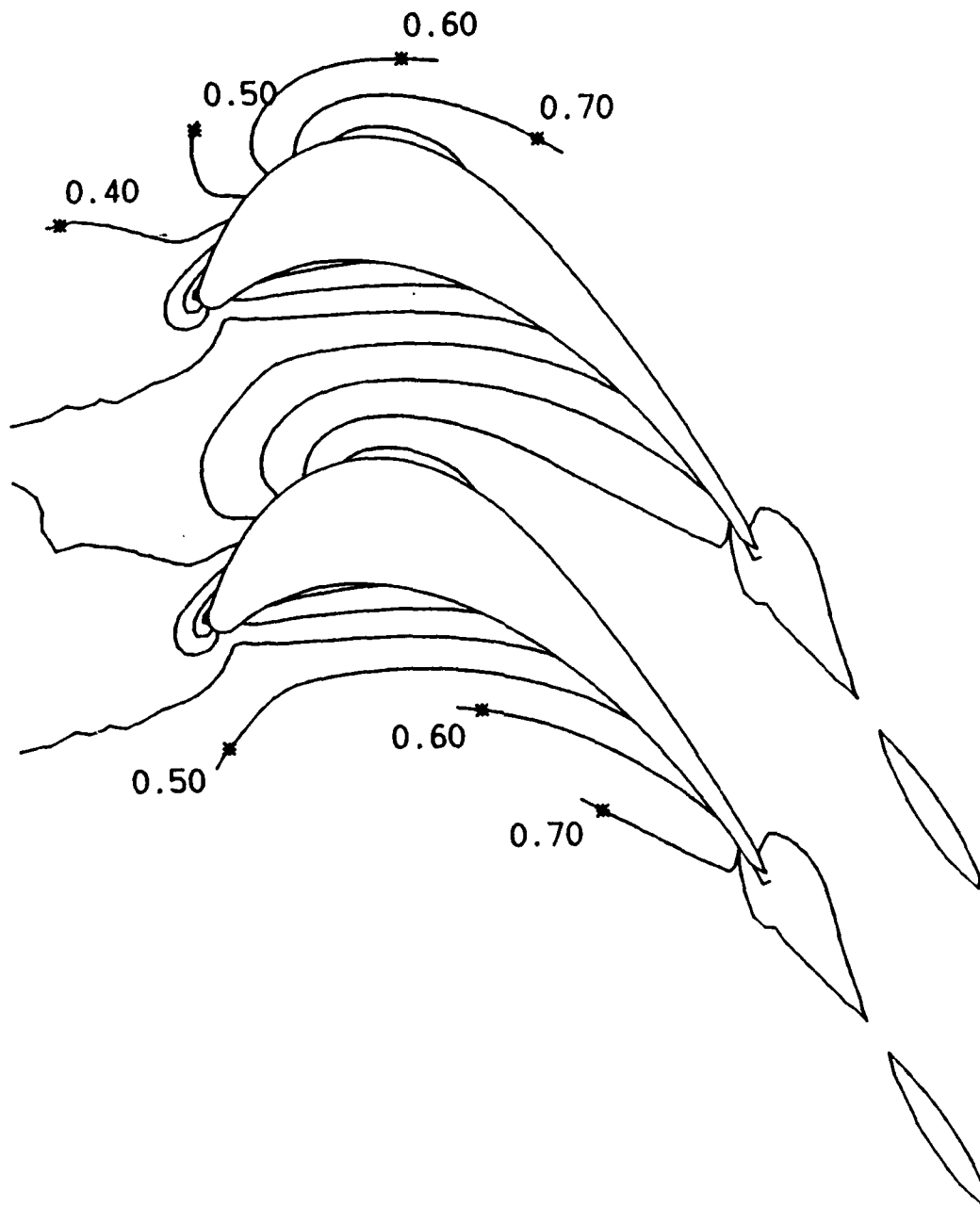


Figure 7-24. Test Case 7. Mach number contours.

VII.1.8 TEST CASE 8

The GMA 400 cascade is considered again in Case 8. The GMA 400 experimental data with the pressure ratio across the blade row which comes closest to matching the pressure ratio of the baseline case is chosen for comparison. That pressure ratio corresponds to a theoretical average exit Mach number of 0.95. For the GMA 400 cascade, this results in a maximum Mach number on the flowfield of approximately 0.96.

Figures 7-25 to 7-27 illustrate the results obtained for Case 8. The experimental and calculated blade surface pressure distribution comparison is very similar to that of Case 7. As shown in Figure 7-25, over most of the blade the agreement between the two is excellent. However, on the pressure side of the blade, where viscous separation occurs in the actual flow, the calculated pressure values are, again, as much as three percent below the measured pressure values.

As Figures 7-25, 7-26, and 7-27 show, the property gradients over the first 30 percent of the suction side of the blade are very high. As with Case 7, these high gradients lower the accuracy of the numerical calculations. This may be the cause of the higher than expected calculated pressure values on that portion of the blade surface. Increasing the grid resolution near the leading edge would improve the resolution. However, the number of grid points was intentionally chosen to provide a reasonable comparison with the grid used for the baseline case.

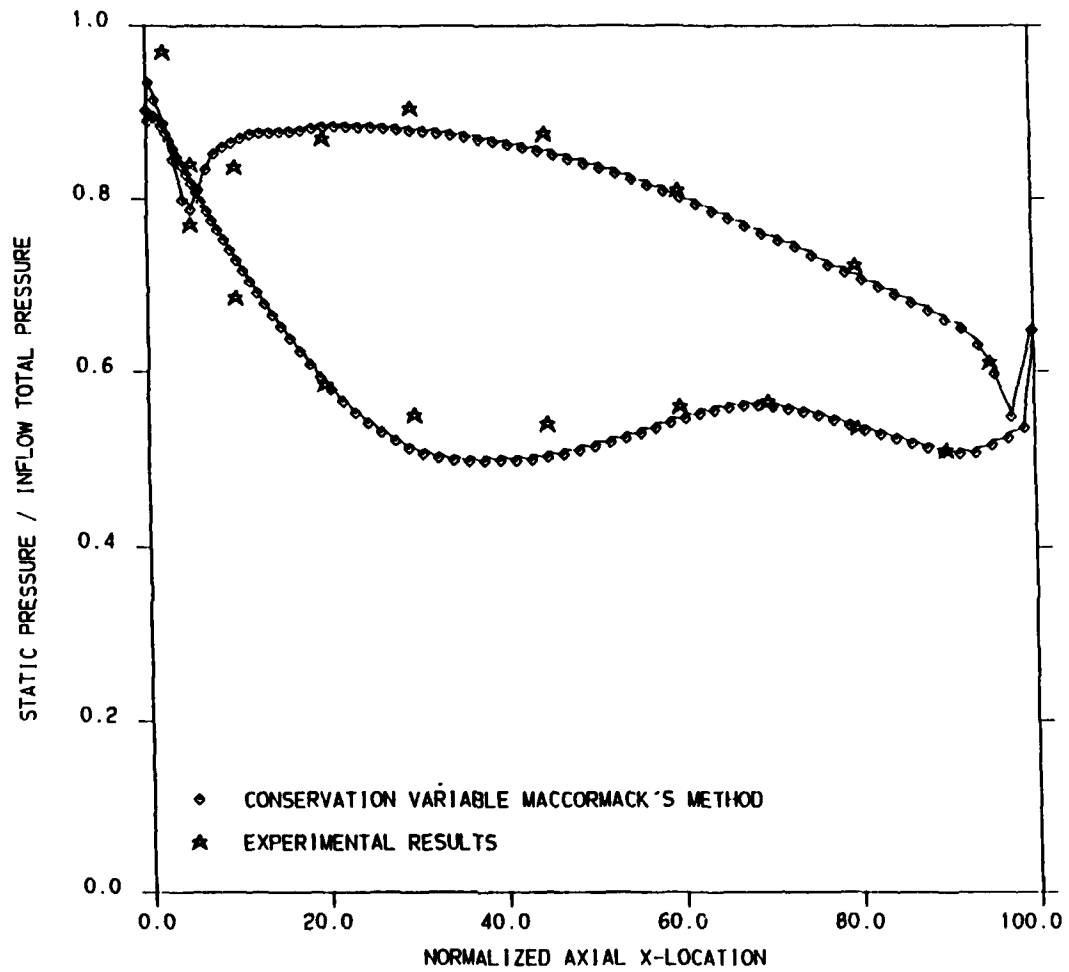


Figure 7-25. Test Case 8. Blade surface normalized static pressure.
GMA 400 cascade, high subsonic flowfield.

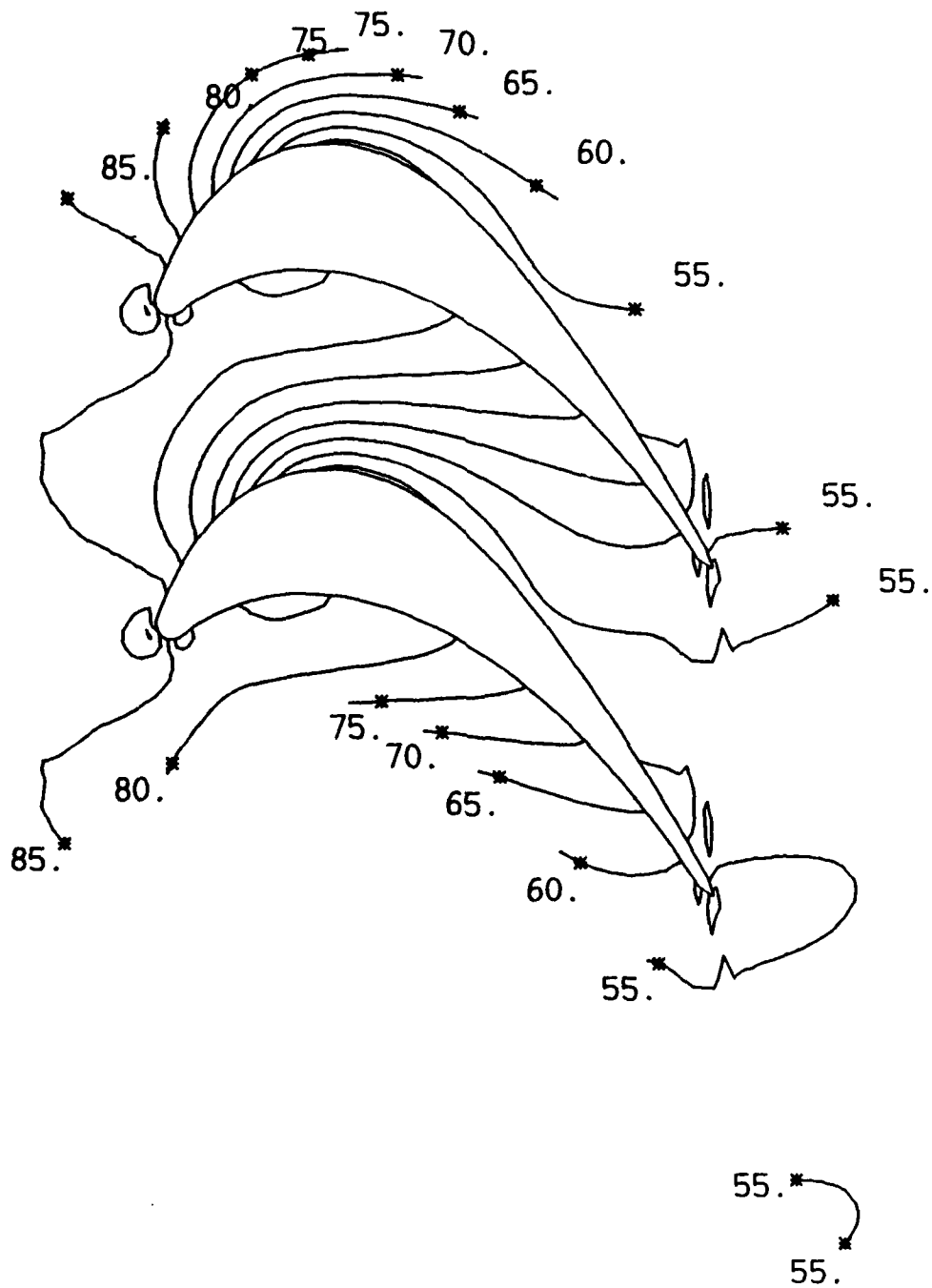


Figure 7-26. Test Case 8. Static pressure contours.

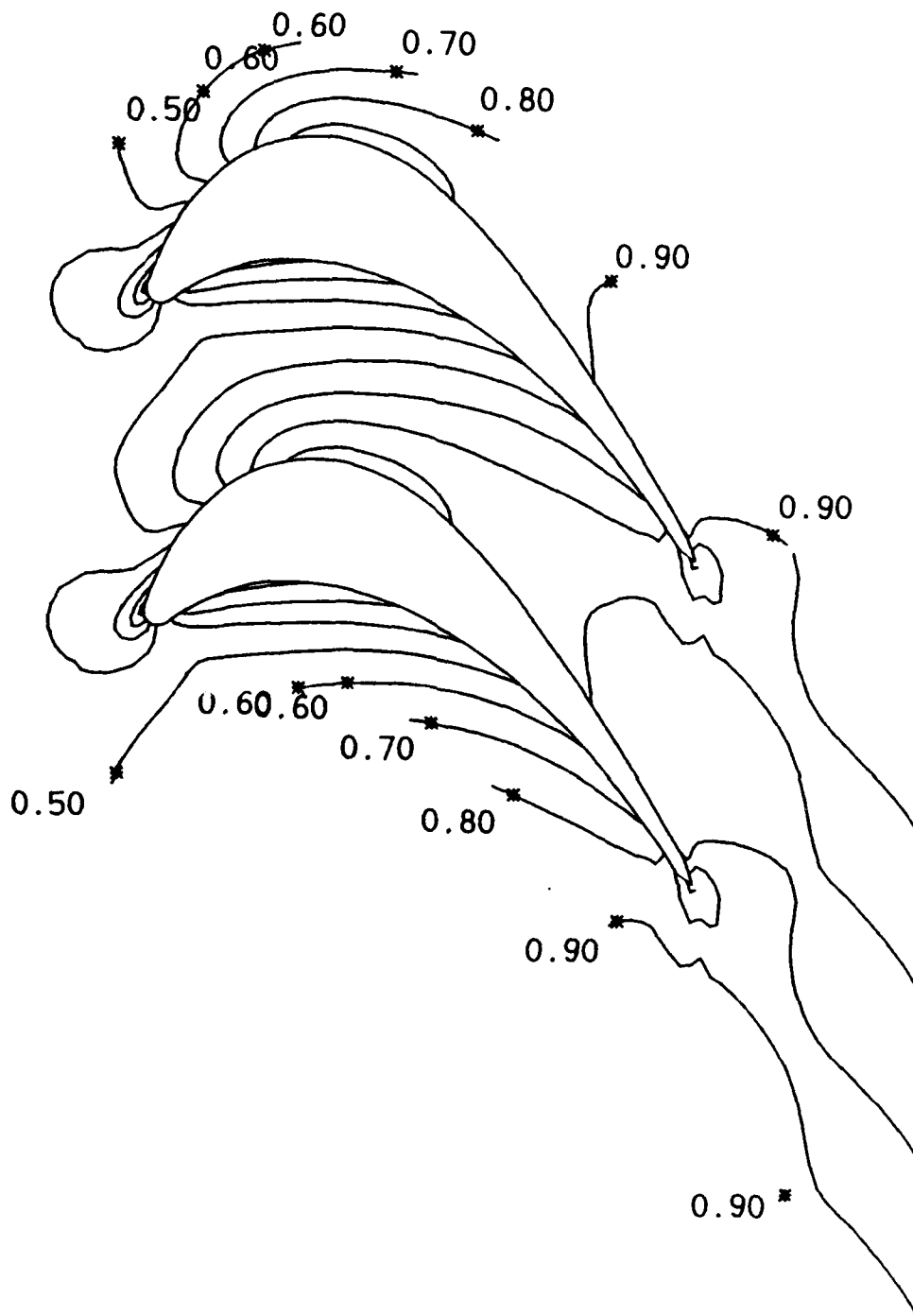


Figure 7-27. Test Case 8. Mach number contours.

VII.2 EXPLICIT ARTIFICIAL DISSIPATION

Experience with the MacCormack code reveals that this combination of methods is slightly unstable, especially at solid wall boundaries. Without explicit damping, the instabilities cause the solution to diverge within a few hundred time steps. The explicit artificial dissipation technique described in Section VI.2.2 is used to control these numerical instabilities.

The impermeable wall boundary condition which is implemented at the blade surface in the cascade flowfield creates the most serious instabilities. Without any damping, instabilities cause very rough property value surfaces in the leading edge region of the blade just prior to local divergence and program termination. When damping is applied in the region at and around the stagnation point, the same sequence of events occurs more slowly at other points along the blade surface. On some grids, when damping is applied at all grid points which are near the blade surface, the same process occurs slowly near the intersection of the periodic boundary and the inflow boundary. For the sake of robustness and computational efficiency, a small amount of smoothing is applied over the entire grid to control these instabilities.

Too much smoothing will either smear the solution excessively, or will itself, cause the solution to diverge. The second derivative terms used to smooth the solution also modify the solution of the equations of motion. If the modification is small compared to the Euler solution itself, the result will reflect the fluid physics. If the solution modification caused by the smoothing is not small compared to the Euler solution, high gradient regions of the

flowfield will be smeared and small scale flow features will be lost. When the effect of the smoothing dominates the Euler solution, the result is unpredictable. In some cases, the large smoothing effect can, itself cause the solution to diverge. Just enough smoothing to control numerical instabilities is optimal.

Since the smoothing does not model any real phenomenon, the correct amount of smoothing can only be determined by experience. Therefore, when a new cascade geometry is solved, or when new flow conditions are input, the appropriate amount of smoothing is usually determined iteratively.

On the baseline case, Case 1, smoothing coefficients were tested, which were 50 percent larger than the iteratively optimized coefficients, to determine the smearing effect of the smoothing scheme. Only very small changes were detected in the final solution. Therefore, the smearing effects of the present smoothing scheme are acceptable.

As described in Section VI.2.2, the smoothing coefficients are linearly decreased from an input maximum value, to an input minimum value. Since the smoothing terms do modify the flow solution, and the magnitude of these terms is decreasing, it can be expected that their changing magnitudes may affect the convergence tolerance check described in Section VI.2.3. Therefore, the amount of smoothing applied is decreased rapidly in order to reach the constant minimum value quickly.

VII.3 TRAILING EDGE POINT

Several things complicate the flow solution at the blade trailing edge. Complications caused by the combination of the blade shape and the inviscid fluid assumption are discussed in this section. The Kutta condition implementation is discussed in the trailing edge point unit process sections of Section IV and Appendix D.

Unlike some compressor blades, turbine blades are not designed to have sharp leading and trailing edges. If turbine blades were manufactured with sharp leading and trailing edges, heat transfer from the very high temperature gases flowing past the blades would over heat these edges.

Unfortunately, rounded trailing edges complicate inviscid flow solutions in two ways. First, as Gostelow [19] discusses, the location on the trailing edge where the suction side and the pressure side flows meet is not obvious. Second, real, viscous flows create boundary layers along the blade surface and a small stagnation region immediately behind the rounded trailing edge. A wake is formed in the flowfield downstream of this aft stagnation point. These features allow the gases to flow smoothly past the rounded trailing edge. This is illustrated in Figure 7-28. Inviscid flow, on the other hand, tries to follow the rounded trailing edge around to the point of application of the Kutta condition. When the flow on each side of the blade reaches the point where the Kutta condition is applied, it must abruptly turn, approximately 90 degrees. This situation is illustrated in Figure 7-29. This is not a realistic situation and is generally not possible numerically.

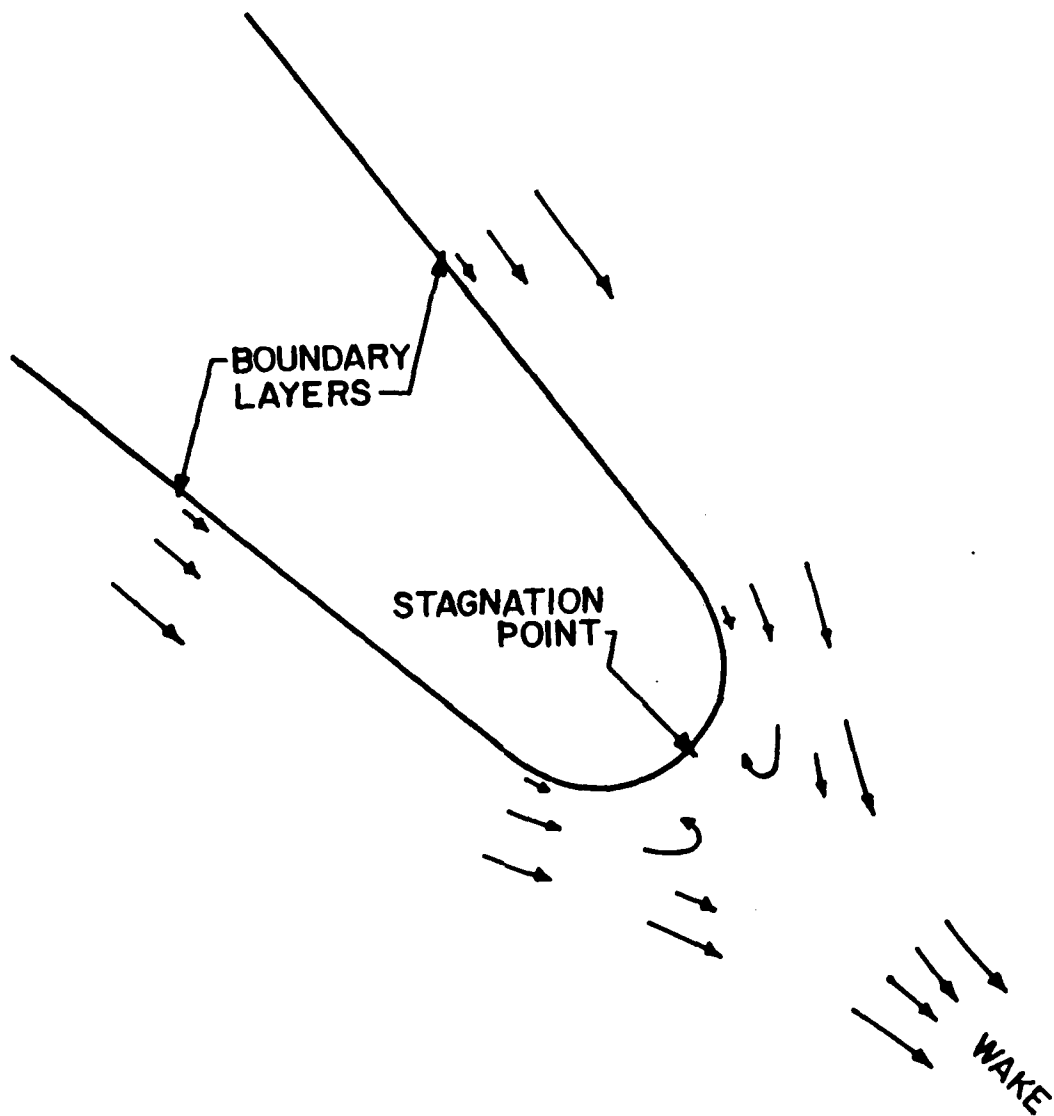


Figure 7-28. Viscous flow past a rounded trailing edge.

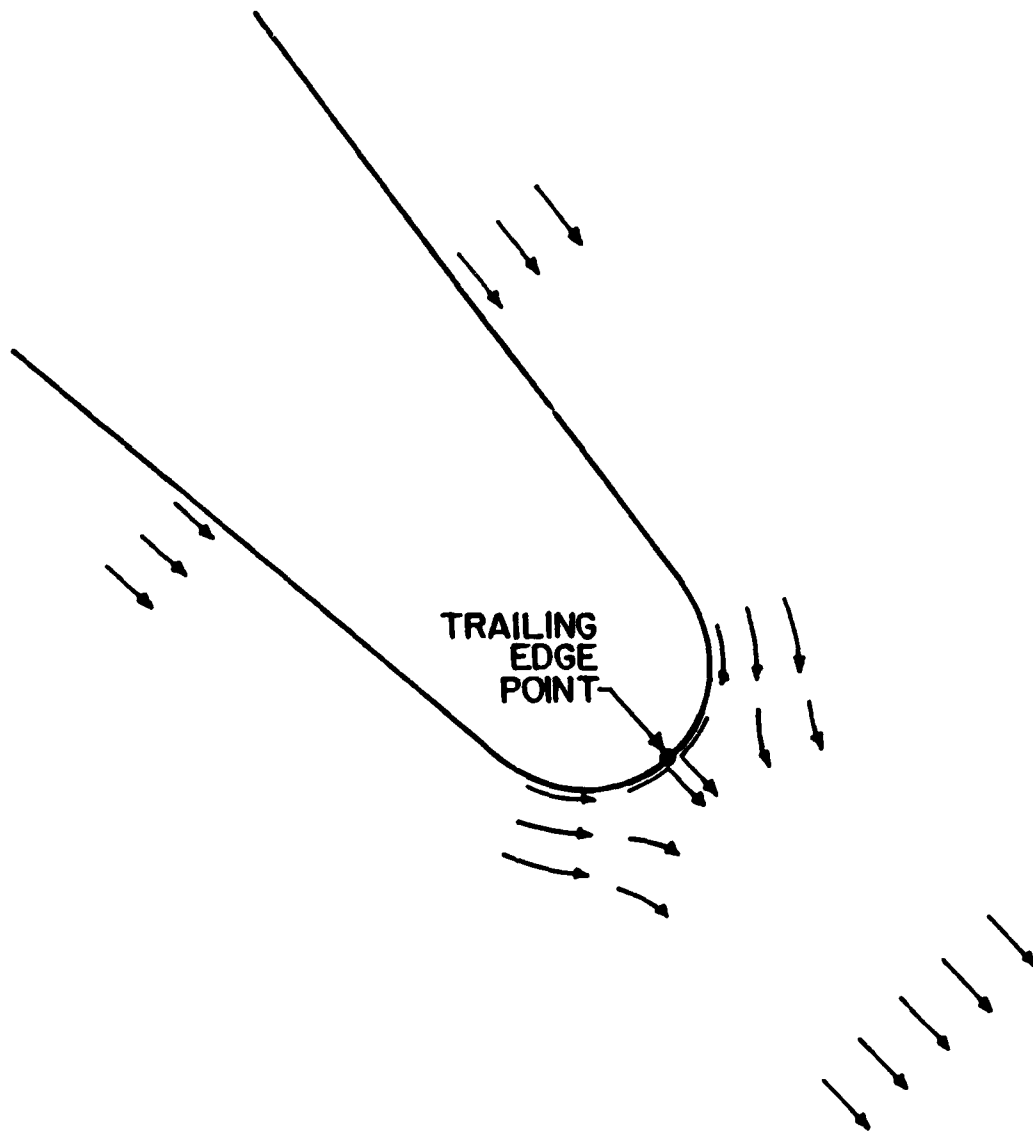


Figure 7-29. Inviscid flow past a rounded trailing edge.

VII.1.7 TEST CASE 7

To demonstrate the capability of the MacCormack code to solve a different cascade flowfield, a distinctly different cascade geometry was chosen. The GMA 400 cascade, illustrated in Figure 7-3, was designed and tested by the Allison Gas Turbine Division of the General Motors corporation [7]. The flowfield through this cascade was calculated at two pressure ratios. Case 7 involves a flowfield with subsonic Mach numbers similar to that of Case 5.

A variety of options for dealing with these problems are available. These options are discussed next.

VII.3.1 TRAILING EDGE POINT LOCATION

Most numerical analysts simply choose a location for the application of the Kutta condition. One of the following locations is commonly chosen.

1. The point where the blade camber line meets the rounded trailing edge.
2. The point where a straight line which is tangent to all the rounded trailing edges in the cascade meets those rounded trailing edges.

Gostelow [19] points out that in reality, this location makes a significant difference in the flow solution around the blade. However, in most numerical solutions, the results at the trailing edge are not reliable. In the present investigation, both of the options listed above were tested on the baseline case, Case 1, and the results were compared. No significant differences were detected. The first of the two options listed above was chosen to produce the grids shown in Figures 7-1 to 7-3.

VII.3.2 BLADE TRAILING EDGE GEOMETRY

The second trailing edge problem discussed above does present serious difficulties to the numerical analyst investigating cascade flows. Three possible solutions are listed below.

1. Modify the shape of the blades being studied so that a sharp trailing edge is produced. This can be accomplished in two manners. The aft portion of the blade can be tapered to a point at the location along the actual rounded trailing edge where the Kutta condition is to be applied. The other technique is to extend the blade trailing edge. This extension is tapered to a point. Property values calculated on the blade extension are ignored.
2. Use relatively large grid point spacing near the trailing edge. This effectively produces the same result as the first part of the previous option.
3. Focus large amounts of smoothing near the trailing edge to damp the instabilities associated with the abrupt turn which occurs at the point of application of the Kutta condition.

Both parts of the first option have two major disadvantages. First, every blade shape which is investigated must be modified. This procedure is both time consuming and arbitrary. Any user of the cascade flow solver must be trained to modify blade shapes prior to using the code. This detracts from the user friendliness of the code. Because the final shape is somewhat arbitrary, different analysts would modify the blades differently. Second, it is difficult to know how the blade shape modifications effect the flow solution.

Two modified AACE II blade shapes were investigated in the present investigation. One blade differed from the actual blade shape over the last ten percent of the blade, the second blade differed from the AACE II blade shape

over only three percent of the blade. Both modified blades produced results very similar to the results gained with the actual AACE II blade. However, this option was discarded because of the requirement to determine new geometries for each blade.

The second option listed above is an automated implementation of the tapered blade shape solution just discussed. Any time a finite number of discrete points are used to represent a curved surface, the shape is modified. The more points that are used, the more accurate the representation. By using relatively sparse grid points in the trailing edge region of a cascade blade, a rounded trailing edge is represented by a point. In this case, analysts need not be trained to modify blade shapes, and the magnitude and arbitrariness of the modification are significantly reduced.

There are disadvantages to the second solution technique. Flow solver truncation errors increase when the point spacing changes abruptly on a finite difference grid. Therefore, by requiring relatively large grid point spacing near the trailing edge, a large portion of the grid is constrained to use a fairly coarse grid. If sufficient accuracy is available from grids of this density, this constraint is not a problem. This solution is implemented in the present investigation.

The third solution described above does not require blade shape modifications and will permit a dense grid. As discussed in Appendix F, the logical way to focus the appropriate amount of additional smoothing in the trailing edge region is to multiply the existing smoothing terms by property

dependent coefficients. These coefficients can be more computationally demanding than the smoothing terms themselves. The added computer time and expense required to compute these coefficients may be prohibitive.

SECTION VIII

CONCLUSIONS

In response to the objectives proposed in Section I, the following conclusions are offered:

1. A robust Poisson-type elliptic grid generator has been developed to generate C-type grids around two-dimensional cascade blades. This code has been used to discretize cascade flowfields for this and other research efforts. The quality of these grids is demonstrated by the high level of accuracy of the flowfield solutions generated on them. The ability of the code to generate C-type grids for cascade blade geometries described by discrete (x,y) input locations, and the ability to handle various stagger angles, blade turning angles, and solidities, all demonstrate the generality of the grid generator.
2. An unsteady inviscid cascade flowfield solver has been developed. The Kentzer method, using conservation variables, has been developed and used to derive an appropriate set of equations to describe the physics at each flowfield boundary. The MacCormack explicit finite difference method is used to numerically integrate these equations at the boundaries and the unsteady Euler equations in the interior of the

flowfield. Unsteady solutions are computed directly with steady flow solutions resulting as asymptotic solutions in time. The MacCormack code has been used to efficiently produce cascade flowfield solutions. Generality of the code has been demonstrated by calculating flowfield solutions for two significantly different cascade geometries and by calculating solutions over the full range of subsonic and transonic Mach numbers typically found in turbines. Robustness has been ensured by incorporating an efficient numerical smoothing scheme.

3. The accuracy of the MacCormack code has been confirmed by demonstrating excellent agreement with experimentally measured blade surface pressure measurements for both of the cascade geometries tested. In addition, a numerical method of characteristics code has been developed, as part of this research effort, to provide a relative standard of accuracy for numerical cascade flowfield solutions. Like the MacCormack code results, solutions generated by the characteristics code were compared to the measured blade surface static pressures. These comparisons prove to be very similar to those generated by the MacCormack code with only very slight improvements in accuracy in regions where the Mach number is close to unity and, on a coarse grid, near the leading edge stagnation region. Static pressure and Mach number contour plots of the flowfield solutions generated by the two codes are also very similar.

LIST OF REFERENCES

LIST OF REFERENCES

1. Steger, J. L. and Sorenson R. L., "Automatic Mesh-Clustering Near A Boundary In Grid Generation with Elliptic Partial Differential Equations," *Journal of Computational Physics*, Vol 33, 1979, pp. 405-410.
2. Kentzer, C. P., "Discretization of Boundary Conditions on Moving Discontinuities," *Second International Conference on Numerical Methods in Fluid Dynamics*, Berkeley, California, September 1970. Published in *Lecture Notes in Physics*, Vol. 8, Springer-Verlag, New York, pp. 108-113.
3. MacCormack. R. W., "The Effect of Velocity in Hypervelocity Impact Cratering," *AIAA Paper 69-354*, April 1969.
4. Marcum, D. L., and Hoffman, J. D., "Calculation of Three-Dimensional Inviscid Flowfields in Propulsive Nozzles with Centerbodies," *AIAA Paper 86-0449*, January 1986.
5. Marcum, D. L., unpublished research, 1986.
6. Huffman, D. G., McClure, R. B., Holtman, R. L., and Sinnet, G. T., "Results of a Two-Dimensional Turbine Cascade (AACE II) Test," *Detroit Diesel Allison Division, General Motors Corp., Research Note 71-47*, July 1971.
7. Smith, R. E., "CAS2D Euler/Thin Layer Navier-Stokes Cascade Code Verification for Transonic Turbine Cascades," *Allison Gas Turbine Division, General Motors Corp.*, August 1987.
8. Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation Foundations and Applications*, North Holland, New York, NY, 1985, pp. 177-179.
9. Vinokur, M. "Conservation Equations of Gas-Dynamics in Curvilinear Coordinate Systems, *Journal of Computational Physics*," Vol 14, 1974, pp. 105-125.
10. Viviand, H., "Conservative Forms of Gas Dynamic Equations," *La Recherche Aerospatiale*, No. 1974-1, 1974, pp. 65-68.

11. Rusanov, V. V., "The Characteristics of General Equations Of Gas Dynamics," *Zhurnal Vychislitelnoi Matematiki Matematicheskoi Fiziki*, Vol. 3, 1963, pp. 508-527. Translated by K. N. Trirogoff, Literature Research Group, Aerospace Library Services, Aerospace Corporation, San Bernardino, California, Report LRG-65-T-38, October 1965.
12. Zucrow, M. J. and Hoffman, J. D., *Gas Dynamics*, Vols. 1 and 2, John Wiley, New York, 1975.
13. Hoffman, J. D., "The Method of Characteristics applied to Unsteady One-, Two-, and Three-Dimensional Flows," Thermal Sciences and Propulsion Center, School of Mechanical Engineering, Purdue University, West Lafayette, Indiana, Report TR-80-07, June 1980.
14. Delaney, R. A. and Kavanagh, P., "Transonic Flow Analysis in Axial-Flow Turbomachinery Cascades by a Time-Dependent Method of Characteristics," ASME Paper 75-GT-8, March 1975.
15. Kemry, M. M., "Computation of Unsteady Two-Dimensional Inviscid Flow in Turbomachinery Cascades," Ph.D. Thesis, Purdue University, May 1981.
16. Delaney, R. A., "Time-Marching Analysis of Steady Transonic Flow in Turbomachinery Cascades Using the Hopscotch Method," ASME Journal of Engineering for Power, Vol. 105, April 1983, pp. 272-279.
17. Srivastava, B. N. and Bozzola, R., "Euler Solutions for Highly Loaded Turbine Cascades," *Journal of Propulsion and Power*, Vol. 3, No. 1, Jan.-Feb., 1987, pp. 39-45.
18. Kwon, O. K., "Navier-Stokes Solution for Steady Two-Dimensional Transonic Cascade Flows," ASME Paper 87-GT-54, June, 1987.
19. Gostelow, J. P., *Cascade Aerodynamics*, Pergamon Press, New York, NY, 1984, pp. 105-106.
20. Akima, H., "A Method of Interpolation and Smooth Curve Fitting Based on Local Procedures," *Journal of the Association for Computing Machinery*, Vol. 17, 1970, pp. 589-602.
21. Chiang, H. D. and Fleeter, S., "Aerodynamic Detuning of a Loaded Airfoil Cascade in an Incompressible Flow by a Locally Analytical Method," Private communication, 1988.
22. Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," *AIAA Journal*, Vol. 24, No. 12, December, 1986, pp. 1931-1940.
23. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

24. Courant, R., Friedrichs, K. O., and Lewy, H., "Über die Partiellen Differenzialgleichungen der Mathematischen Physik," *Mathematische Annalen*, Vol. 100, 1928, pp. 32-74.
25. Wang, B. N., and Hoffman, J. D., "Calculation of Annular Nozzle Trisomic Flowfields by the Method of Characteristics," *AIAA Paper 86-1519*, June 1986.
26. Marcum, D. L., and Hoffman, J. D., "Calculation of Three-Dimensional Flowfields by the Unsteady Method of Characteristics," *AIAA Journal*, Vol. 23, No. 10, October 1985, pp. 1497-1505.
27. Butler, D. S., "The Numerical Solution of Hyperbolic Systems of Partial Differential Equations in Three Independent Variables," *Proceedings of the Royal Society of London*, Vol. 255 A, 1960, pp. 232-252.

APPENDICES

APPENDIX A

GOVERNING EQUATIONS AND COORDINATE TRANSFORMATIONS

Appendix A describes the governing equations for the current investigation. The conservation variable forms of the governing equations are provided with subsequent expansion to primitive variables. In addition, the equations are presented in matrix form and are transformed into strong conservation form.

A.1 GOVERNING EQUATIONS IN CONSERVATION VARIABLE FORM

As stated in Section III, the equations applicable to this research effort are the continuity equation, the vector momentum equation, the energy equation, and the thermal and caloric equations of state. These equations are repeated here.

$$(\rho)_t + \nabla \cdot (\rho \vec{V}) = 0 \quad (\text{A.1})$$

$$(\rho \vec{V})_t + \nabla \cdot (\rho \vec{V} \vec{V}) + \nabla P = 0 \quad (\text{A.2})$$

$$(\rho e)_t + \nabla \cdot [\vec{V}(\rho e + P)] = 0 \quad (\text{A.3})$$

$$T = \frac{P}{\rho R} \quad (\text{A.4})$$

$$a = \left[\frac{\gamma P}{\rho} \right]^{\frac{1}{2}} \quad (\text{A.5})$$

These equations are based on the following assumptions:

1. continuum flow,
2. inviscid flow,
3. no body forces,
4. no heat conduction,
5. simple thermodynamic system, and
6. thermally and calorically perfect gas.

A.2 EXPANSION TO PRIMITIVE FORM

In the present investigation the governing partial differential equations of motion are used in both primitive and conservation variable forms. This section presents the expansion of the Euler equations, equations (A.1) to (A.3), from conservation variable form, to primitive variable form.

A.2.1 CONTINUITY EQUATION. The conservation variable form of the continuity equation, equation (A.1), is expanded to primitive variables in the following manner:

$$\rho_t + \vec{V} \cdot (\nabla \rho) + \rho(\nabla \cdot \vec{V}) = 0 \quad (\text{A.6})$$

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) = 0 \quad (\text{A.7})$$

A.2.2 VECTOR MOMENTUM EQUATION. The conservation variable form of the vector momentum equation, equation (A.2), contains the divergence of the fluid density times the dyad of the velocity vector, $\nabla \cdot (\rho \bar{V} \bar{V})$. This term represents the divergence of the convection of momentum. This term is better understood when it is expanded in the following manner:

First, expand one of the velocity vectors, $\bar{V} = u\hat{i} + v\hat{j}$, to obtain:

$$\nabla \cdot (\rho \bar{V} \bar{V}) = \nabla \cdot (\rho \bar{V} u \hat{i} + \rho \bar{V} v \hat{j}) \quad (\text{A.8})$$

Next, perform the indicated divergence operation:

$$\nabla \cdot (\rho \bar{V} \bar{V}) = (\rho u \bar{V})_x + (\rho v \bar{V})_y \quad (\text{A.9})$$

Then, by expanding the remaining velocity vector into its components and rearranging, the familiar convective momentum terms appear.

$$\nabla \cdot (\rho \bar{V} \bar{V}) = [(\rho u^2)_x + (\rho uv)_y] \hat{i} + [(\rho uv)_x + (\rho v^2)_y] \hat{j} \quad (\text{A.10})$$

Returning to vector notation, this term is expanded into primitive variable form in the following manner:

$$\nabla \cdot (\rho \bar{V} \bar{V}) = \bar{V} [\nabla \cdot (\rho \bar{V})] + [(\rho \bar{V}) \cdot \nabla] \bar{V} \quad (\text{A.11})$$

$$\nabla \cdot (\rho \bar{V} \bar{V}) = \rho \bar{V} (\nabla \cdot \bar{V}) + \bar{V} [\bar{V} \cdot (\nabla \rho)] + [(\rho \bar{V}) \cdot \nabla] \bar{V} \quad (\text{A.12})$$

Using equation (A.12), the conservation variable form of the vector momentum equation, equation (A.2), is expanded to primitive variable form in the following manner:

$$\bar{V} \rho_t + \rho \bar{V}_t + \rho \bar{V} (\nabla \cdot \bar{V}) + \bar{V} [\bar{V} \cdot (\nabla \rho)] + [(\rho \bar{V}) \cdot \nabla] \bar{V} + \nabla P = 0 \quad (\text{A.13})$$

Combining appropriate terms into substantial derivatives and rearranging yields:

$$\vec{V} \frac{D\rho}{Dt} + \rho \frac{D\vec{V}}{Dt} + \rho \vec{V}(\nabla \cdot \vec{V}) + \nabla P = 0 \quad (\text{A.14})$$

Equation (A.14) can be simplified further. The first and third terms of equation (A.14) combine to form the velocity times the continuity equation, equation (A.7). Therefore, by substituting the continuity equation and dividing by the density, equation (A.14) reduces to:

$$\rho \frac{D\vec{V}}{Dt} + \nabla P = 0 \quad (\text{A.15})$$

A.2.3 ENERGY EQUATION. Given the assumptions listed at the beginning of this section, the total energy, ρe , can be expressed in several different forms. First, in terms of internal energy and kinetic energy:

$$\rho e = \rho \hat{u} + \rho \frac{V^2}{2} \quad (\text{A.16})$$

Next, assuming a calorically perfect gas:

$$\rho e = \rho C_v T + \rho \frac{V^2}{2} \quad (\text{A.17})$$

and, assuming a thermally perfect gas:

$$\rho e = \frac{P}{\gamma - 1} + \rho \frac{V^2}{2} \quad (\text{A.18})$$

These relationships will be used in the expansion of the energy equation, equation (A.3), into primitive variables. First, expand all terms except the ρe product:

$$(\rho e)_t + \vec{V} \cdot \nabla(\rho e) + (\rho e)(\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P + P(\nabla \cdot \vec{V}) = 0 \quad (A.19)$$

Substituting equation (A.18) into equation (A.19) yields:

$$\begin{aligned} & \left[\frac{P}{\gamma-1} + \rho \frac{V^2}{2} \right]_t + \vec{V} \cdot \nabla \left[\frac{P}{\gamma-1} + \rho \frac{V^2}{2} \right] \\ & + \left[\frac{P}{\gamma-1} + \rho \frac{V^2}{2} \right] (\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P + P(\nabla \cdot \vec{V}) = 0 \end{aligned} \quad (A.20)$$

Combining terms into substantial derivatives where possible results in a primitive variable form of the energy equation:

$$\begin{aligned} & \frac{1}{\gamma-1} \frac{DP}{Dt} + \left[\frac{V^2}{2} \right] \frac{D\rho}{Dt} + \rho \frac{D \left[\frac{V^2}{2} \right]}{Dt} \\ & + \frac{\gamma P}{\gamma-1} (\nabla \cdot \vec{V}) + \rho \frac{V^2}{2} (\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P = 0 \end{aligned} \quad (A.21)$$

This form of the energy equation is used in Appendix C to demonstrate the equivalence of the primitive variable and the conservation variable forms of the compatibility equations.

Another common primitive variable form of the energy equation which incorporates the assumptions applicable to this project is derived next. Expand equation (A.3) in the following manner:

$$e\rho_t + \rho e_t + \rho \vec{V} \cdot \nabla e + e \vec{V} \cdot \nabla \rho + \rho e \nabla \cdot \vec{V} + P \nabla \cdot \vec{V} + \vec{V} \cdot \nabla P = 0 \quad (A.22)$$

Combining the appropriate terms into substantial derivatives where possible and using the continuity equation, equation (A.7), to simplify equation (A.22) yields:

$$\rho \frac{De}{Dt} + P \nabla \cdot \vec{V} + \vec{V} \cdot \nabla P = 0 \quad (\text{A.23})$$

Using the momentum equation, equation (A.15), the pressure gradient term in equation (A.23) can be exchanged for the following convective term:

$$\vec{V} \cdot \nabla P = - \rho \vec{V} \cdot \left[\frac{D\vec{V}}{Dt} \right] \quad (\text{A.24})$$

Since $\vec{V} \cdot \frac{D\vec{V}}{Dt} = \frac{D}{Dt} \left[\frac{V^2}{2} \right]$, equation (A.23) becomes:

$$\rho \frac{De}{Dt} + P \nabla \cdot \vec{V} = \rho \frac{D}{Dt} \left[\frac{V^2}{2} \right] \quad (\text{A.25})$$

Expanding the energy, e , into internal and kinetic energy:

$$\rho \frac{D(\hat{u} + \frac{V^2}{2})}{Dt} + P \nabla \cdot \vec{V} = \rho \frac{D}{Dt} \left[\frac{V^2}{2} \right] \quad (\text{A.26})$$

Canceling equal and opposite terms and dividing by the density leaves:

$$\rho \frac{D\hat{u}}{Dt} + P \nabla \cdot \vec{V} = 0 \quad (\text{A.27})$$

Using the definition of specific enthalpy, $h = \hat{u} + P/\rho$:

$$\frac{Dh}{Dt} = \frac{D\hat{u}}{Dt} + \frac{D(P/\rho)}{Dt} = \frac{D\hat{u}}{Dt} + \frac{1}{\rho} \frac{DP}{Dt} - \frac{P}{\rho^2} \frac{D\rho}{Dt} \quad (\text{A.28})$$

Solving equation (A.28) for the substantial derivative of internal energy, \hat{u} , provides a substitution for the first term in equation (A.27):

$$\frac{Dh}{Dt} - \frac{1}{\rho} \frac{DP}{Dt} + \frac{P}{\rho^2} \frac{D\rho}{Dt} + \frac{P}{\rho} \nabla \cdot \vec{V} = 0 \quad (\text{A.29})$$

The primitive variable continuity equation, equation (A.7), reveals that the last two terms in equation (A.29) sum to zero. Therefore,

$$\frac{Dh}{Dt} - \frac{1}{\rho} \frac{DP}{Dt} = 0 \quad (\text{A.30})$$

For a thermally and calorically perfect gas:

$$h = C_p T = \frac{\gamma}{\gamma-1} \frac{P}{\rho} \quad (\text{A.31})$$

Therefore,

$$\frac{Dh}{Dt} = \frac{\gamma}{\gamma-1} \left[\frac{1}{\rho} \frac{DP}{Dt} - \frac{P}{\rho^2} \frac{D\rho}{Dt} \right]$$

Using this equation to modify equation (A.30) yields:

$$\frac{\gamma}{\gamma-1} \left[\frac{1}{\rho} \frac{DP}{Dt} - \frac{P}{\rho^2} \frac{D\rho}{Dt} \right] - \frac{1}{\rho} \frac{DP}{Dt} = 0 \quad (\text{A.32})$$

Rearranging simplifies equation (A.32).

$$\frac{DP}{Dt} - \frac{\gamma P}{\rho} \frac{D\rho}{Dt} = 0 \quad (\text{A.33})$$

For a perfect gas, $a^2 = \gamma P / \rho$. Therefore, the primitive variable energy equation becomes:

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (\text{A.34})$$

Thus, for the assumptions listed above, this primitive variable form of the energy equation is equivalent to equations (A.3) and (A.21). Equation (A.34) is

the form of the energy equation used in the method of characteristics.

A.3 GOVERNING EQUATIONS IN SHORTHAND NOTATION

The conservation variable form of the governing equations can be presented in the following short hand notation:

$$(\rho)_t = \mathcal{C} \quad (\text{A.35})$$

$$(\rho \vec{V})_t = \vec{\mathcal{M}} \quad (\text{A.36})$$

$$(\rho e)_t = \mathcal{E} \quad (\text{A.37})$$

where \mathcal{C} , $\vec{\mathcal{M}}$ and \mathcal{E} contain the space derivatives appearing in the continuity, vector momentum, and energy equations, respectively. Written in vector notation, \mathcal{C} , $\vec{\mathcal{M}}$, and \mathcal{E} are:

$$\mathcal{C} = -\nabla \cdot (\rho \vec{V}) \quad (\text{A.38})$$

$$\vec{\mathcal{M}} = -\nabla \cdot (\rho \vec{V} \vec{V}) - \nabla P \quad (\text{A.39})$$

$$\mathcal{E} = -\nabla \cdot [\vec{V}(\rho e + P)] \quad (\text{A.40})$$

A.4 GOVERNING EQUATIONS IN MATRIX FORM

The matrix form of the governing equations is:

$$Q_t + E_x + F_y = 0 \quad (\text{A.41})$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad E = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (\rho e + P)u \end{bmatrix} \quad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (\rho e + P)v \end{bmatrix}$$

A.5 COORDINATE TRANSFORMATIONS

Coordinate transformations are often utilized in numerical computations to improve solution accuracy. This requires two major steps: the transformation of the grid covering the physical domain of interest to a uniform orthogonal computational grid, and the transformation of the governing equations from the physical domain to the computational domain. The grid transformation is addressed in Section IV and in Appendix E. The transformation of the governing equations is presented in this section.

Since neither the physical nor the computational grids is time dependent in the present investigation, derivatives with respect to time are unaffected by the transformation. Therefore, only space transformations are considered in this section. The physical and transformed spaces are related through a one-to-one correspondence. That is, each point in the physical domain, (x,y) , is related to exactly one point in the computational domain, (ξ,η) , and vice versa. Therefore, the computational space is described by:

$$\begin{aligned}\xi &= \xi(x,y) \\ \eta &= \eta(x,y)\end{aligned}\tag{A.42}$$

or conversely

$$\begin{aligned}x &= x(\xi,\eta) \\ y &= y(\xi,\eta)\end{aligned}\tag{A.43}$$

Partial derivatives of the generic variable, $f = f(x,y)$, are transformed using the partial differentiation chain rule:

$$f_x = f_\xi \xi_x + f_\eta \eta_x \quad (\text{A.44})$$

$$f_y = f_\xi \xi_y + f_\eta \eta_y \quad (\text{A.45})$$

The terms ξ_x , ξ_y , η_x , and η_y are the transformation metrics. To calculate these metrics, the generic function f in equation (A.44) is replaced by x and y , respectively, to form the following identities:

$$x_x = x_\xi \xi_x + x_\eta \eta_x = 1$$

$$y_x = y_\xi \xi_x + y_\eta \eta_x = 0$$

or, in matrix form:

$$\begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} \xi_x \\ \eta_x \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (\text{A.46})$$

Cramer's rule provides a straightforward solution to this system of equations.

Thus,

$$\xi_x = J y_\eta \quad (\text{A.47})$$

$$\eta_x = -J y_\xi \quad (\text{A.48})$$

where J is the Jacobian of the transformation. By definition J is given by the following determinant:

$$J = \begin{vmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{vmatrix} \quad (\text{A.49})$$

This determinant is equivalent to the reciprocal of the Jacobian of the inverse transformation, I . In other words, $J = I^{-1}$, where

$$I = \begin{bmatrix} x_{\xi} & x_{\eta} \\ y_{\xi} & y_{\eta} \end{bmatrix} \quad (\text{A.50})$$

The inverse Jacobian, I , naturally arises when Cramer's rule is used to solve equation (A.46). Physically, the Jacobian represents a stretching function which relates a differential volume in physical space to that same volume in the transformed space.

A procedure parallel to that presented above, using equation (A.45), yields the remaining transformation metrics:

$$\xi_y = -J x_{\eta} \quad (\text{A.51})$$

$$\eta_y = J x_{\xi} \quad (\text{A.52})$$

Since the computational grid in the transformed space is orthogonal and equally spaced, all the inverse metrics (i.e., x_{ξ} , x_{η} , y_{ξ} , and y_{η}) can be calculated numerically to a high order of accuracy. Careful thought should be given to the method of calculation of the transformation metrics. This topic is discussed further in Section A.6.

A.5.1 TRANSFORMED GOVERNING EQUATIONS. Transforming the space derivatives in equation (A.41) as shown in equations (A.44) and (A.45) results in the following system of equations in weak conservation form:

$$Q_t + \xi_x E_{\xi} + \eta_x E_{\eta} + \xi_y F_{\xi} + \eta_y F_{\eta} = 0 \quad (\text{A.53})$$

Equation (A.53) can also be cast in strong conservation form. In the strong conservation form, the transformation metrics in equation (A.53), which are coefficients of the derivatives of E and F , are placed inside the differential

operators. For example, consider the following partial derivative:

$$[IE\xi_x]_\xi \quad (A.54)$$

Expanding this partial derivative by the chain rule yields:

$$[IE\xi_x]_\xi = E_\xi[I\xi_x] + E[I\xi_x]_\xi \quad (A.55)$$

The first term on the right-hand side of equation (A.55) is the same as the second term in equation (A.53), after multiplication by the inverse Jacobian. Solving equation (A.55) for this term yields:

$$E_\xi[I\xi_x] = [IE\xi_x]_\xi - E[I\xi_x]_\xi \quad (A.56)$$

Similar equalities can be formed for each of the terms in equation (A.53). Making the appropriate substitutions and rearranging yields the following result:

$$\begin{aligned} [IQ]_t + [I(E\xi_x + F\xi_y)]_\xi + [I(E\eta_x + F\eta_y)]_\eta \\ - E [(I\xi_x)_\xi + (I\eta_x)_\eta] - F [(I\xi_y)_\xi + (I\eta_y)_\eta] = 0 \end{aligned} \quad (A.57)$$

Substituting equations (A.47), (A.48), (A.51), and (A.52) into the last two bracketed terms of equation (A.57) shows that each of these terms is exactly zero. Therefore, equation (A.57) reduces to:

$$[IQ]_t + [I(E\xi_x + F\xi_y)]_\xi + [I(E\eta_x + F\eta_y)]_\eta = 0 \quad (A.58)$$

This form of the governing equations, known as the strong conservation-law form, was first developed by Vinokur [9] and Viviand [10].

A.5.2 CONTRAVARIANT VELOCITIES. Equation (A.58) contains several groupings of velocity products of the following form:

$$U = \xi_x u + \xi_y v \quad (\text{A.59})$$

$$V = \eta_x u + \eta_y v \quad (\text{A.60})$$

These groupings are called contravariant velocities. Contravariant velocities represent components of the transformed velocity in the directions of the computational coordinates. For example, U represents the velocity component in the ξ direction. This becomes obvious in a Lagrangian frame when, for example, U is calculated by taking the total derivatives of ξ with respect to time. Thus,

$$U = \frac{d\xi}{dt} = \xi_x \frac{dx}{dt} + \xi_y \frac{dy}{dt} \quad (\text{A.61})$$

where, by definition, $\frac{dx}{dt} = u$ and $\frac{dy}{dt} = v$, are the Cartesian velocity components. Substituting these contravariant velocities into the transformed governing equations improves computational efficiency. Thus, the strong conservation-law governing equations, with the contravariant velocities incorporated, are:

$$\bar{Q}_t + \bar{E}_\xi + \bar{F}_\eta = 0 \quad (\text{A.62})$$

where

$$\bar{Q} = I \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad \bar{E} = \begin{bmatrix} I(\rho U) \\ I(\rho u U + \xi_x P) \\ I(\rho v U + \xi_y P) \\ I[(\rho e + P)U] \end{bmatrix} \quad \bar{F} = \begin{bmatrix} I(\rho V) \\ I(\rho u V + \eta_x P) \\ I(\rho v V + \eta_y P) \\ I[(\rho e + P)V] \end{bmatrix}$$

In this research program, equation (A.62) is solved using the MacCormack explicit finite difference method [3] to calculate properties at all points in the interior of the flowfield. The Kentzer method [2] yields similar sets of equations applicable at the various boundary points.

A.6 CALCULATION OF TRANSFORMATION METRICS

Caution should be exercised when calculating the transformation metrics. Thompson et al. [8] explain that the highest level of accuracy is obtained when the metrics are differenced numerically in exactly the same manner as the dependent variables in the governing equations. For example, if a partial derivative in the governing equations is approximated by a central difference, then any metric used in conjunction with that derivative should also be approximated by a central difference using exactly the same grid points. This statement applies for equations cast in primitive variables as well as in conservation variables. As described in Section IV, the MacCormack method calculates all derivatives using one-sided differences. Therefore, the same one-sided differences must be used to calculate the metrics.

To understand this precaution more thoroughly, consider the two terms in equation (A.57) which were determined to be exactly zero and were therefore, discarded. For illustrative purposes, consider the first of these terms:

$$E [(I\xi_x)_\xi + (I\eta_x)_\eta] \quad (\text{A.63})$$

The metrics in this term were calculated using equations (A.47) and (A.48).

Substituting these metric identities into equation (A.63) yields:

$$E [(IJy_\eta)_\xi + (I(-J)y_\xi)_\eta] \quad (\text{A.64})$$

which can be written as:

$$E [(y_\eta)_\xi - (y_\xi)_\eta] \quad (\text{A.65})$$

The MacCormack method differentiates the y_η and y_ξ metrics using forward one-sided differences for the predictor and backward one-sided differences for the corrector. Considering the predictor:

$$E \left[\left\{ y_{\eta_{i+1,j}} - y_{\eta_{i,j}} \right\} - \left\{ y_{\xi_{i,j+1}} - y_{\xi_{i,j}} \right\} \right] \quad (\text{A.66})$$

where $\Delta\xi = \Delta\eta = 1.0$. First, consider the case where the predictor metrics have also been calculated using forward one-sided differences. For example, $y_\xi = y_{i+1,j} - y_{i,j}$ and $y_\eta = y_{i,j+1} - y_{i,j}$, etc. Inserting these approximations into equation (A.66) yields:

$$E \left[\left(\left\{ y_{i+1,j+1} - y_{i+1,j} \right\} - \left\{ y_{i,j+1} - y_{i,j} \right\} \right) - \left(\left\{ y_{i+1,j+1} - y_{i,j+1} \right\} - \left\{ y_{i+1,j} - y_{i,j} \right\} \right) \right] \quad (\text{A.67})$$

Obviously, this term is exactly zero. The same result is obtained when backward differences are used consistently.

Next, consider a technique which does not use a consistent approximation for metric derivatives. For this test case, use centered differences to approximate the inverse metrics. For example, $y_\xi = [y_{i+1,j} - y_{i-1,j}]/2$ and $y_\eta = [y_{i,j+1} - y_{i,j-1}]/2$, etc. Inserting these approximations into equation (A.66) yields:

$$\begin{aligned} \frac{E}{2} \left[\left(\left\{ y_{i+1,j+1} - y_{i+1,j-1} \right\} - \left\{ y_{i,j+1} - y_{i,j-1} \right\} \right) \right. \\ \left. - \left(\left\{ y_{i+1,j+1} - y_{i-1,j+1} \right\} - \left\{ y_{i+1,j} - y_{i-1,j} \right\} \right) \right] \quad (A.68) \end{aligned}$$

In general, this term does not equal zero! Therefore, by not using consistent approximations, the terms which were discarded because they are equal to zero analytically, are in fact, not equal to zero numerically. In other words, an undesirable source term is added to the transformed equations when the derivative approximations used to calculate inverse metrics are different from the approximations used to calculate property derivatives. The net effect after both predictor and corrector steps is proportional to the difference between current time level properties and the predicted new time level properties (i.e., $E^{n+1} - \overline{E^{n+1}}$). If this difference is exactly zero, the source term will also be zero. However, this is seldom the case. Even if exact calculations are used to calculate inverse metrics, the numerical approximation of the second derivative, which is calculated using the MacCormack method, is not exact and will still create an error.

In order to avoid the undesirable source terms described above, the approximation used to calculate the inverse metrics must be the same approximation used to calculate flow property derivatives. In the case of the MacCormack method, different approximations are used at the predictor and corrector steps. Therefore, two sets of metrics must be used. One set must be calculated using first-order forward differences and the other set must be calculated using first-order backward differences.

Close examination reveals another interesting point. Only one set of Jacobian values is required. The inverse metrics discussed above are multiplied by the local Jacobian to calculate the metrics. Then, in the transformation to the strong conservation form, all terms are divided by the local Jacobian. Therefore, in every term containing one metric (all first-order space derivatives), the effect of the Jacobian is eliminated. However, the Jacobian in the time derivatives in the strong conservation form is not cancelled out and is not affected by the consistency considerations described above. Therefore, the most accurate approximation for the Jacobian should be used.

APPENDIX B

METHOD OF CHARACTERISTICS

Characteristic theory identifies the physical paths of propagation of information through a flowfield, which are the pathline and the wave surfaces. The numerical method of characteristics uses characteristic theory to numerically integrate the governing differential equations of motion. This appendix describes the results of characteristic theory for the Euler equations and outlines the numerical method of characteristics approach chosen as a method of verification of the MacCormack code in the present investigation.

B.1 CHARACTERISTIC THEORY

Detailed developments of the method of characteristics have been published by Rusanov [11], Zucrow and Hoffman [12], Hoffman [13], and Wang [25]. This section summarizes the development as presented by Wang.

The governing equations for the present investigation are discussed in Appendix A. These equations form a set of hyperbolic partial differential equations of the first-order. In primitive variable form they are:

$$\rho_t + \vec{V} \cdot \nabla \rho + \rho \nabla \cdot \vec{V} = 0 \quad (\text{B.1})$$

$$\rho \frac{D\vec{V}}{Dt} + \nabla P = 0 \quad (\text{B.2})$$

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (B.3)$$

The method of characteristics forms linear combinations of this set of equations yielding an equivalent set of compatibility equations. A compatibility equation is an interior operator which has one less independent variable, contains derivatives only in the corresponding characteristic surface, and is valid only in that surface. A characteristic surface is a surface in the solution space on which the governing partial differential equations may be combined linearly to form a compatibility equation.

For unsteady two-dimensional flow there are three independent variables: (x, y, t) . Therefore, the characteristic surfaces are surfaces in three-dimensional space. There are two families of characteristic surfaces: stream surfaces and wave surfaces. There are two types of characteristic curves corresponding to the two families of characteristic surfaces: pathlines and wavelines. The pathline is the intersection of all the stream surfaces containing the given point in space which is under consideration. The waveline is the line of contact between a wave surface and the Mach conoid. The Mach conoid is the envelope of all wave surfaces which contain the given point in space which is under consideration. Two stream surfaces and the corresponding pathline are illustrated in Figure B-1. A Mach conoid, a wave surface, and the corresponding waveline are illustrated in Figure B-2.

There are two types of compatibility equations corresponding to the two types of characteristic surfaces: the pathline equation and the waveline equation. Compatibility equations contain directional derivatives only along a

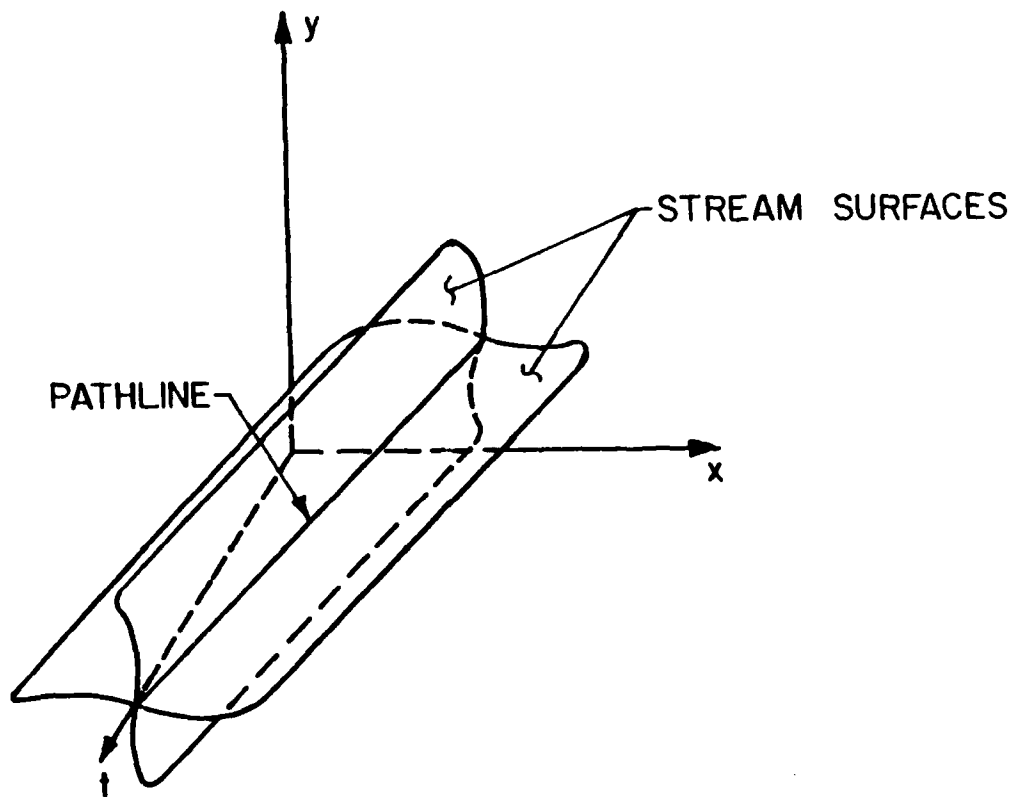


Figure B-1. Stream surfaces and corresponding pathline.

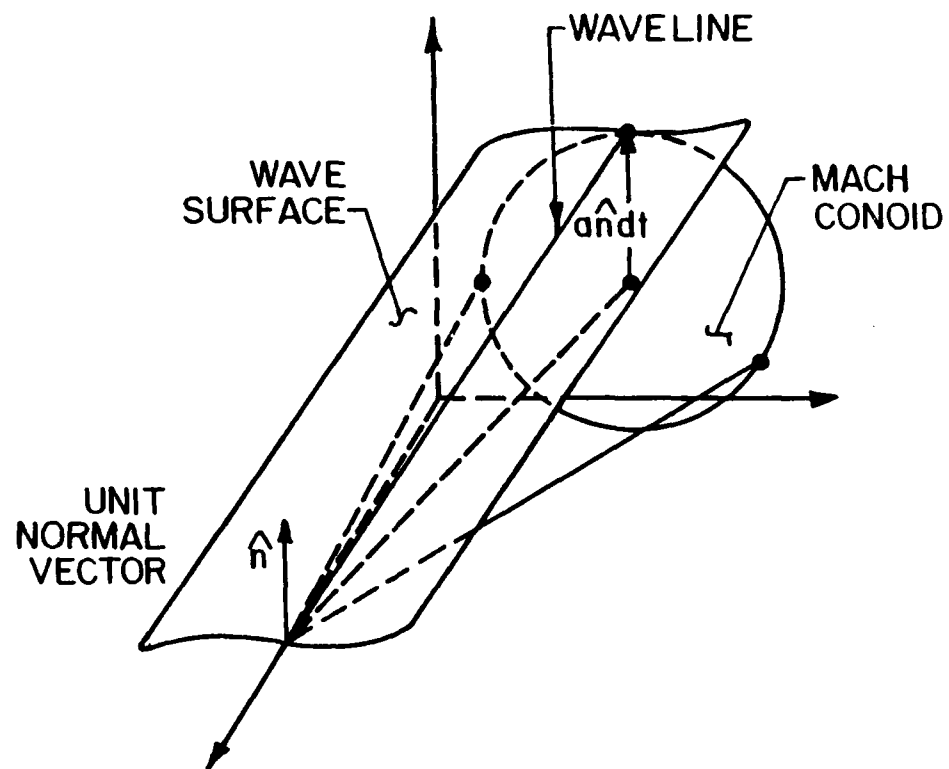


Figure B-2. Mach conoid, wave surface and corresponding waveline.

characteristic curve and in the corresponding surface. The directional derivative along a characteristic curve is given by:

$$\left. \frac{d(\)}{dt} \right|_{\text{curve}} = (\)_t + \vec{W} \cdot \nabla (\) \quad (\text{B.4})$$

where \vec{W} is the relative velocity vector along the curve.

Since the relative velocity along a pathline is the fluid velocity, the directional derivative along the pathline is:

$$\left. \frac{d(\)}{dt} \right|_{\text{pathline}} = \frac{D(\)}{Dt} = (\)_t + \vec{V} \cdot \nabla (\) \quad (\text{B.5})$$

which is the substantial derivative. The energy equation in primitive variable form, equation (B.3), is a valid compatibility equation as it contains directional derivatives only along the pathline. Therefore, the primitive variable form of the pathline equation is:

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (\text{B.6})$$

The relative velocity along a waveline is:

$$\vec{W} = \vec{V} - a\hat{n} \quad (\text{B.7})$$

where \hat{n} is the unit vector normal to the corresponding wave hypersurface. Thus, the directional derivative along a waveline is:

$$\left. \frac{d(\)}{dt} \right|_{\text{waveline}} = \frac{\mathcal{D}(\)}{\mathcal{D}(\)} = (\)_t + (\vec{V} - a\hat{n}) \cdot \nabla (\) \quad (\text{B.8})$$

The primitive variable governing equations, equations (B.1), (B.2), and (B.3), can be combined in the following manner to obtain a valid waveline equation:

$$a^2 \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{continuity} \\ \text{equation} \end{bmatrix} - \rho a \hat{n} \cdot \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{momentum} \\ \text{equation} \end{bmatrix} + \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{energy} \\ \text{equation} \end{bmatrix} = 0 \quad (\text{B.9})$$

Inserting equations (B.1) to (B.3) into equation (B.9) and rearranging to form directional derivatives along the wavelines where possible, forms the following waveline compatibility equation:

$$\frac{\partial P}{\partial t} - \rho a \hat{n} \cdot \frac{\partial \vec{V}}{\partial t} + \rho a^2 \left[\nabla \cdot \vec{V} - \hat{n} \cdot (\hat{n} \cdot \nabla) \vec{V} \right] = 0 \quad (\text{B.10})$$

The term in brackets contains derivatives in the corresponding wave surface which are called cross derivatives.

The pathline equation, equation (B.6), is valid on the pathline. The waveline equation, equation (B.10), is valid on the wave surface corresponding to the wave surface unit normal vector, \hat{n} . There are an infinite number of choices for the wave surface unit normal vector \hat{n} corresponding to the infinite number of wave surfaces at a point. However, only four compatibility equations can be included as a set of independent equations for unsteady two-dimensional flow. The pathline equation is included since it is the only compatibility equation which contains a derivative of density. Consequently, exactly three waveline equations are used.

B.2 THE NUMERICAL METHOD OF CHARACTERISTICS

Because the numerical method of characteristics is founded on characteristic theory, it very closely models the fluid physics of the flowfield being studied, and therefore, is very accurate. Unfortunately, it is also

complicated and computationally demanding. Marcum and Hoffman [26] have developed a numerical method of characteristics analysis which is second-order accurate in time and space and is very efficient relative to other numerical method of characteristics analyses. In a manner similar to the implementation of Marcum and Hoffman, the present investigation employs the numerical method of characteristics to solve unsteady two-dimensional cascade flowfields.

In the numerical method of characteristics the computational coordinates are the characteristic surfaces described in Section B.1. Therefore, numerical method of characteristics analyses integrate the compatibility equations along the corresponding characteristic lines. In direct marching numerical method of characteristics analyses this is accomplished by constructing the network of characteristic lines, which make up the computational grid, as the flowfield solution is computed. This process is very computationally demanding in flowfields involving more than two independent variables.

The inverse marching numerical method of characteristics uses a fixed grid to prespecify solution point locations at the forward time level and projects a local network of characteristic lines from those solution points back to the current time level solution surface. Interpolation is used to determine property values at the points where the characteristic lines intersect the current time level solution surface. This inverse marching technique is much more efficient than the direct marching technique for flowfields involving three or more independent variables such as the unsteady two-dimensional cascade flowfield being studied in the present investigation.

B.2.1 APPROACH. In the present investigation an inverse marching numerical method of characteristics scheme is employed. The solution points are the grid points of the AACE II cascade grids which are also used by the MacCormack code. These grids are illustrated in Figures 7-1, and 7-2, and their development is discussed in Section V.

Successive solution surfaces are separated by time steps which are restricted by the CFL [24] stability criterion. The CFL criterion requires the physical domain of dependence be contained within the numerical domain of dependence. Figure B-3 illustrates the numerical domain of dependence, or convex hull, for the inverse marching numerical method of characteristics. As described in Section VI.2.1 for the MacCormack code, local or global time steps can be used in the characteristics code.

The second-order accurate modified-Euler predictor-corrector numerical integration method is used to integrate the compatibility equations. For the predictor step of the modified-Euler method, the derivatives along the characteristic lines are replaced by one-sided forward-difference approximations. Cross derivatives and coefficients are evaluated at the current time level. For the corrector step, the derivatives along the characteristic lines are replaced by finite difference approximations centered along the characteristic lines. Coefficients are evaluated as averages between their values at the current time level and their predicted values at the forward time level. In order to maintain second-order accuracy, without violating the physical domain of dependence, evaluation of the cross derivative terms at the forward time level is avoided in the manner developed by Butler [27].

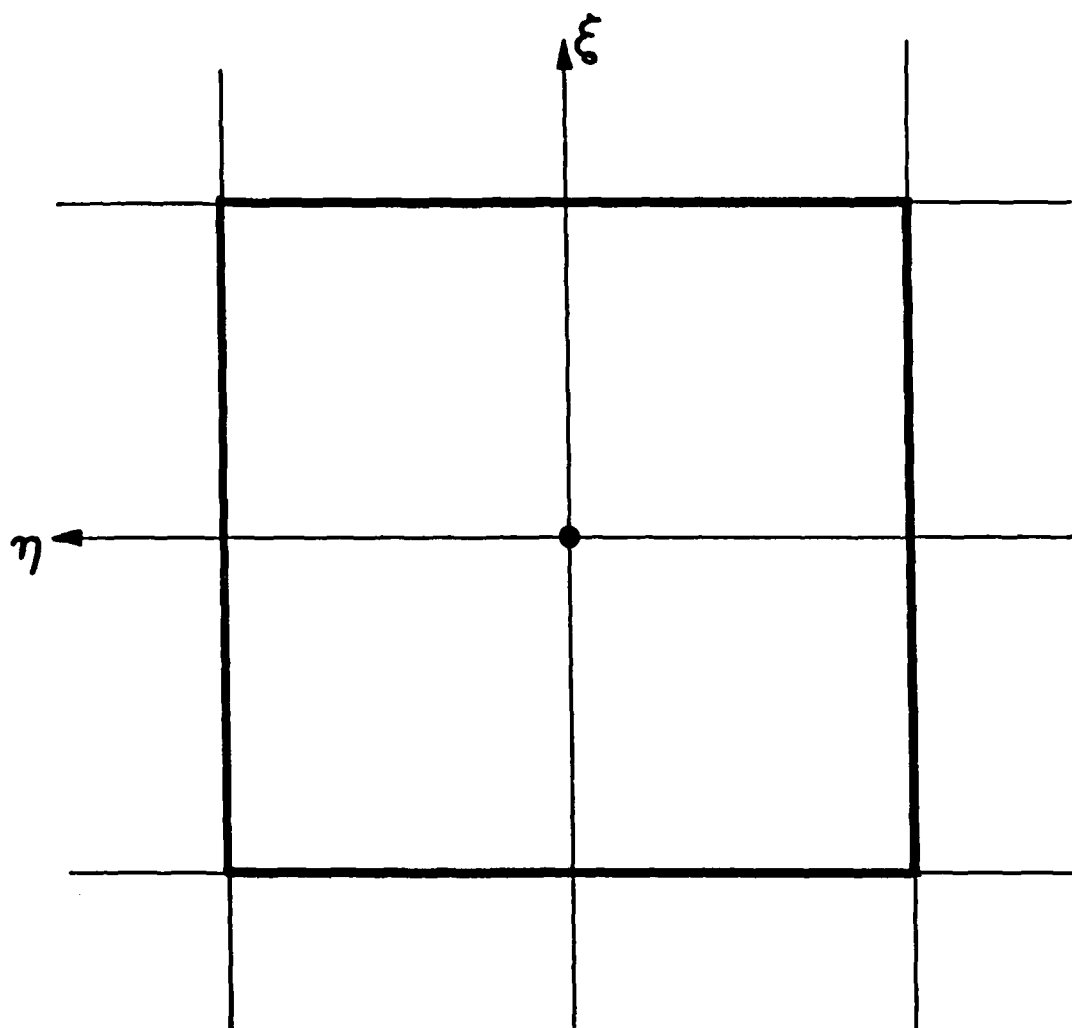


Figure B-3. Convex hull for the numerical method of characteristics.

Butler devised a scheme that explicitly eliminates the need to evaluate cross derivatives at the forward time level. Butler replaced the set of three independent waveline equations by a set of four waveline equations and a noncharacteristic linear combination of the continuity equation and the energy equation. The noncharacteristic relation is:

$$\frac{DP}{Dt} + \rho a^2 \Delta \cdot \vec{V} = 0 \quad (B.11)$$

By forming three linear combinations of the finite difference forms of these five equations, the cross derivative terms at the forward time solution point are eliminated. These three equations, along with the pathline equation, comprise a set of four independent finite difference equations for determining the four primitive variable solution properties, ρ , P , u , and v , at the forward time level solution point. Figure B-4 illustrates the pathline and four wavelines between two solution surfaces. An indexing system for the local characteristic network is also presented in Figure B-4. In that system, the points where the wavelines intersect the current time level solution surface are numbered 1, 2, 3, and 4, the point where the pathline intersects the current time level solution surface is numbered 5, and the solution point at the forward time level solution surface is numbered 6.

B.2.2 INTERPOLATION. Interpolation is used to determine property values at the points where the the characteristic lines intersect the current time level surface, points 1 to 5. Since the physical grid is nonorthogonal and nonequally spaced, second-order accurate interpolation of each of the solution properties,

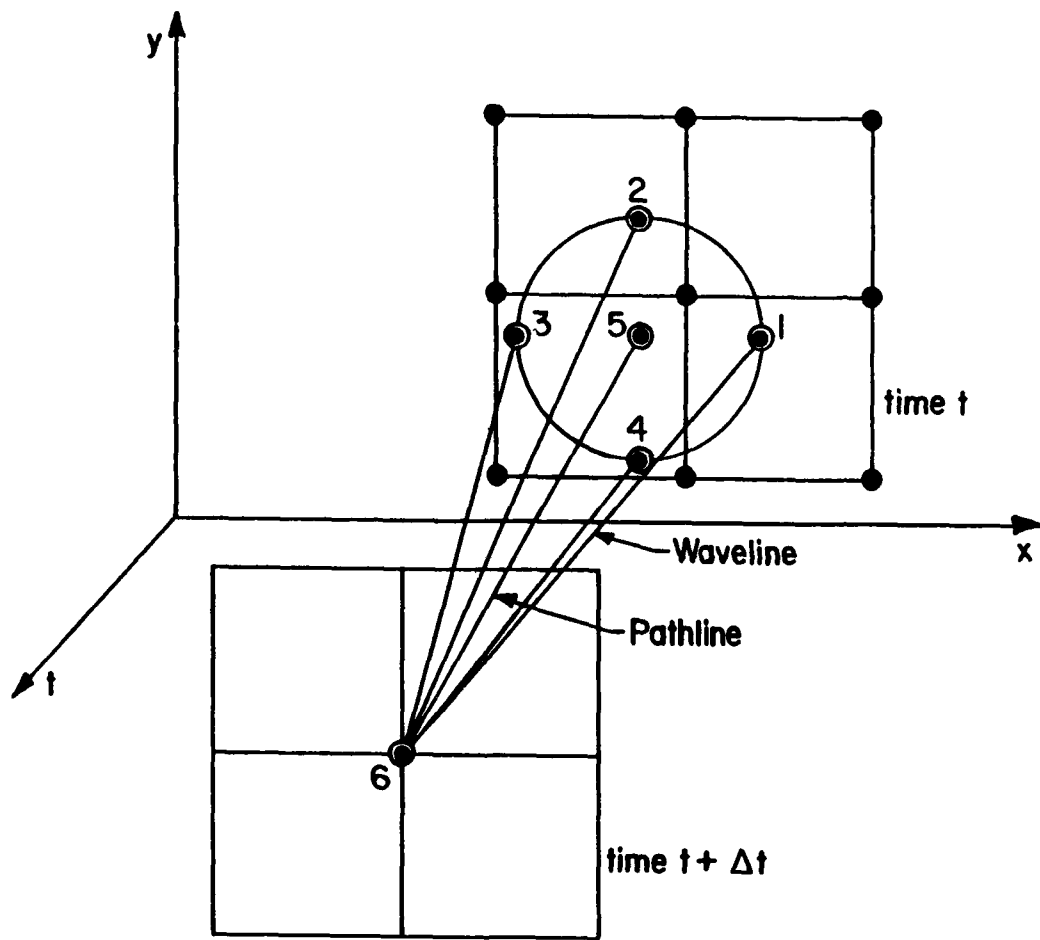


Figure B-4. Pathline, four wavelines, and two solution surfaces.

at every solution point, at every time level can be very computationally demanding. However, efficiency is greatly improved by transforming the grid to an equally spaced orthogonal grid and interpolating in transformed space. Grid transformation is discussed in Appendix E, and the transformation of partial differential equations from physical to computational space is discussed in Appendix A.

Flow property values in the current time level solution surface are determined by least squares quadratic bivariate curve fits in transformed space of the data at the nine initial-value grid points which lie within the convex hull illustrated in Figure B-3. The quadratic bivariate interpolating polynomial for the flow properties in transformed interpolating space is:

$$f = a_0 + a_1 \Delta\xi + a_2 \Delta\eta + a_3 \Delta\xi \Delta\eta + a_4 \Delta\xi^2 + a_5 \Delta\eta^2 \quad (\text{B.12})$$

where f is any of the four solution property values, ρ , P , u , or v , and $\Delta\xi$ and $\Delta\eta$ are the transformed distances from points 1, 2, 3, 4, or 5 to the central point of the interpolating square. Specifically:

$$\Delta\xi = \xi_x \Delta x + \xi_y \Delta y \quad (\text{B.13})$$

and

$$\Delta\eta = \eta_x \Delta x + \eta_y \Delta y \quad (\text{B.14})$$

where Δx and Δy are the distances in physical space from points 1, 2, 3, 4, or 5 to the central point of the interpolating square. The values of Δx and Δy are obtained from the finite difference forms of the pathline equation:

$$\Delta x_5 = -u_{65} \Delta t \quad (\text{B.15})$$

$$\Delta y_5 = -v_{65} \Delta t \quad (\text{B.16})$$

or of the waveline equation:

$$\Delta x_m = -(u - a n_i)_{6m} \Delta t \quad (\text{B.17})$$

$$\Delta y_m = -(v - a n_j)_{6m} \Delta t \quad (\text{B.18})$$

where the subscript m represents any of the waveline base points, points 1, 2, 3, or 4, and the double subscripts indicate average values along the appropriate characteristic line. For the predictor step these average coefficient values are replaced by the corresponding values at the current time level.

The coefficients a_0 to a_5 in equation (B.12) are determined through a least squares procedure applied in transformed space. In matrix form:

$$A a = b \quad (\text{B.19})$$

where a is the six component vector of coefficients a_0 to a_5 , b is the six component vector of terms which consist of summations over the nine grid points in the interpolating square of products of f , $\Delta\xi$, and $\Delta\eta$, and A is the six by six matrix of coefficients involving summation of products of $\Delta\xi$ and $\Delta\eta$. In the b vector and the A matrix, $\Delta\xi$ and $\Delta\eta$ are the distances in transformed space from the central point to each of the points in the interpolating square. Since $\Delta\xi = \Delta\eta = 1.0$ over the entire grid, the A matrix is the same for every interpolating square. This unique A matrix also has a unique inverse. Therefore, A^{-1} is multiplied times the b vector to solve for the a vector explicitly at every grid point at every time step.

transformation metrics at the nine points on the same transformed interpolating square used for flow property interpolations. The bivariate interpolating polynomial is again equation (B.12) where f is one of the four transformation metrics, ξ_x , ξ_y , η_x , or η_y . Because the grid is fixed, the coefficients for the transformation metrics interpolating polynomials, a_0 to a_5 , are calculated once, before the first time step, and used at every time step.

B.2.3 UNIT PROCESSES. In many ways the unit processes developed for this characteristics code are similar to the MacCormack code unit processes presented in Section IV. For example, all the same types of boundary points must be considered and the boundary conditions implemented are the same. On the other hand, some considerations differ from those of the MacCormack code. For example, primitive variables are used instead of conservation variables, each solution point is predicted and corrected prior to moving to the next solution point, and the numerical method of characteristics is employed, rather than the MacCormack method.

Each of the following types of flowfield points must be considered:

1. interior points,
2. solid wall boundary points,
3. trailing edge point,
4. subsonic inlet boundary points,
5. subsonic exit boundary points, and

6. supersonic exit boundary points.

All periodic boundary points and grid cut points lie in one of the regions listed above. Therefore, no additional sets of equations are required to compute these points. However, because of their location on the computational grid, they do require special handling of indices.

All unit processes have several features in common. At every solution point the following sequence is followed:

1. For the predictor:
 - a. Interpolating polynomial coefficients are determined for the four solution properties, ρ , P , u , and v .
 - b. The physical locations of any of the five base points, points 1 to 5, required for the unit process under consideration, are calculated using the characteristic equations, equations (B.15) to (B.18).
 - c. The transformed space locations of any of the five base points, points 1 to 5, required for the unit process under consideration, are calculated using equations (B.13) and (B.14).
 - d. Using the appropriate interpolating polynomial coefficients, a_0 to a_5 , in conjunction with the interpolating polynomial, equation (B.12), each of the solution properties, ρ , P , u , and v , are determined at any of the five base points, points 1 to 5, required for the unit process under consideration.
 - e. Using the appropriate interpolating polynomial coefficients, a_0 to

a_5 , in conjunction with equations (B.12), (B.22), and (B.23), the transformation metrics, ξ_x , ξ_y , η_x , and η_y , and the cross derivative terms, u_x , u_y , v_x , and v_y , are determined at any of the four waveline base points, points 1 to 4, required for the unit process under consideration.

- f. Using the equations presented below for the type of point being considered, the four solution properties are predicted.
2. For the corrector many of the same steps are executed. However, in order to achieve second-order accuracy, the values of coefficients appearing in the equations are averaged between the values at the base points, points 1 to 5, and the predicted values at the solution point, point 6. For the corrector step:
 - a. The physical locations of any of the five base points, points 1 to 5, required for the unit process under consideration, are recalculated using the characteristic equations, equations (B.15) to (B.18).
 - b. The transformed space locations of any of the five base points, points 1 to 5, required for the unit process under consideration, are recalculated using equations (B.13) and (B.14).
 - c. Using the appropriate interpolating polynomial coefficients, a_0 to a_5 , in conjunction with the interpolating polynomial, equation (B.12), each of the solution properties, ρ , P , u , and v , are determined at any of the five base points, points 1 to 5, required for the unit process under consideration.

- d. Using the appropriate interpolating polynomial coefficients, a_0 to a_5 , in conjunction with equations (B.12), (B.22), and (B.23), the transformation metrics, ξ_x , ξ_y , η_x , and η_y , and the cross derivative terms, u_x , u_y , v_x , and v_y , are determined at any of the four waveline base points, points 1 to 4, required for the unit process under consideration.
- e. Using the equations presented below for the type of point being considered, the four solution properties are corrected.

B.2.3.1 INTERIOR POINT UNIT PROCESS. At interior points the pathline equation, equation (B.6), is solved to determine the density at the solution point, point 6. The finite difference form of the pathline equation is:

$$\rho_6 = \rho_5 + \frac{1}{a_{56}^2} [P_6 - P_5] \quad (\text{B.25})$$

where the double subscripts indicate average values along the appropriate characteristic line, which in this case is the pathline.

The remaining solution properties, P , u , and v , are determined using the method developed by Butler [27]. Butler used four waveline equations, equation (B.10), and the noncharacteristic relation, equation (B.11), to produce three independent equations. The particular linear combination used by Butler eliminates the terms involving cross derivatives at the forward time level. The unit vectors associated with the four waveline equations can not be arbitrarily specified. Specifically, these unit vectors must be chosen in orthogonal pairs where $\hat{n}_1 = -\hat{n}_3$, $\hat{n}_2 = -\hat{n}_4$ and $\hat{n}_1 \times \hat{n}_2 = \hat{t}$.

The finite difference form of the noncharacteristic relation is:

$$P_6 + \frac{\Delta t}{2} (\rho a^2)_6 \left[u_x + v_y \right]_6 = C_5 \quad (\text{B.26})$$

where

$$C_5 = P_5 - \frac{\Delta t}{2} (\rho a^2)_5 \left[u_x + v_y \right]_5$$

The four waveline equations in finite difference form are:

$$P_6 - (\rho a)_{61} (n_{1i} u_6 + n_{1j} v_6) - \frac{\Delta t}{2} \left\{ \rho a^2 [(n_{1i}^2 - 1) u_x + (n_{1j}^2 - 1) v_y + n_{1i} n_{1j} (u_y + v_x)] \right\}_6 = C_1 \quad (\text{B.27})$$

where

$$C_1 = P_1 - (\rho a)_{61} (n_{1i} u_1 + n_{1j} v_1) + \frac{\Delta t}{2} \left\{ \rho a^2 [(n_{1i}^2 - 1) u_x + (n_{1j}^2 - 1) v_y + n_{1i} n_{1j} (u_y + v_x)] \right\}_1$$

$$P_6 - (\rho a)_{62} (n_{2i} u_6 + n_{2j} v_6) - \frac{\Delta t}{2} \left\{ \rho a^2 [(n_{2i}^2 - 1) u_x + (n_{2j}^2 - 1) v_y + n_{2i} n_{2j} (u_y + v_x)] \right\}_6 = C_2 \quad (\text{B.28})$$

where

$$C_2 = P_2 - (\rho a)_{62} (n_{2i} u_2 + n_{2j} v_2) + \frac{\Delta t}{2} \left\{ \rho a^2 [(n_{2i}^2 - 1) u_x + (n_{2j}^2 - 1) v_y + n_{2i} n_{2j} (u_y + v_x)] \right\}_2$$

$$P_6 - (\rho a)_{63} (n_{3i} u_6 + n_{3j} v_6)$$

$$- \frac{\Delta t}{2} \left[\rho a^2 [(n_{3i}^2 - 1) u_x + (n_{3j}^2 - 1) v_y + n_{3i} n_{3j} (u_y + v_x)] \right]_6 = C_3 \quad (B.29)$$

where

$$C_3 = P_3 - (\rho a)_{63} (n_{3i} u_3 + n_{3j} v_3)$$

$$+ \frac{\Delta t}{2} \left[\rho a^2 [(n_{3i}^2 - 1) u_x + (n_{3j}^2 - 1) v_y + n_{3i} n_{3j} (u_y + v_x)] \right]_3$$

$$P_6 - (\rho a)_{64} (n_{4i} u_6 + n_{4j} v_6)$$

$$- \frac{\Delta t}{2} \left[\rho a^2 [(n_{4i}^2 - 1) u_x + (n_{4j}^2 - 1) v_y + n_{4i} n_{4j} (u_y + v_x)] \right]_6 = C_4 \quad (B.30)$$

where

$$C_4 = P_4 - (\rho a)_{64} (n_{4i} u_4 + n_{4j} v_4)$$

$$+ \frac{\Delta t}{2} \left[\rho a^2 [(n_{4i}^2 - 1) u_x + (n_{4j}^2 - 1) v_y + n_{4i} n_{4j} (u_y + v_x)] \right]_4$$

The noncharacteristic relation and the four waveline equations are combined to form three independent equations in the following manner.

Subtracting equation (B.27) from equation (B.29) yields:

$$\left[(\rho a)_{63} + (\rho a)_{61} \right] (n_{1i} u_6 + n_{1j} v_6) = C_3 - C_1 \quad (B.31)$$

Similarly, subtracting equation (B.28) from equation (B.30) yields:

$$\left[(\rho a)_{64} + (\rho a)_{62} \right] (n_{2i} u_6 + n_{2j} v_6) = C_4 - C_2 \quad (B.32)$$

The third equation is derived by subtracting the noncharacteristic relation from the sum of the four waveline equations divided by two:

$$\begin{aligned} P_6 + \frac{1}{2} \left\{ \left[(\rho a)_{63} - (\rho a)_{61} \right] (n_{1i} u_6 + n_{1j} v_6) + \left[(\rho a)_{64} - (\rho a)_{62} \right] (n_{2i} u_6 + n_{2j} v_6) \right\} \\ = \frac{1}{2} (C_1 + C_2 + C_3 + C_4) - C_5 \end{aligned} \quad (B.33)$$

At interior points equations (B.31) to (B.33) can be simplified by choosing $\hat{n}_1 = \hat{i}$ and $\hat{n}_2 = \hat{j}$. This forces $\hat{n}_3 = -\hat{i}$ and $\hat{n}_4 = -\hat{j}$. Using these unit vectors and rearranging equations (B.31) to (B.33) yields:

$$u_6 = \frac{[C_3 - C_1]}{[(\rho a)_{63} + (\rho a)_{61}]} \quad (B.34)$$

$$v_6 = \frac{[C_4 - C_2]}{[(\rho a)_{64} + (\rho a)_{62}]} \quad (B.35)$$

$$P_6 = \frac{1}{2} \left\{ u_6 \left[(\rho a)_{61} - (\rho a)_{63} \right] + v_6 \left[(\rho a)_{62} - (\rho a)_{64} \right] + C_1 + C_2 + C_3 + C_4 \right\} - C_5 \quad (B.36)$$

Equations (B.25) and (B.34) to (B.36) are solved to advance the solution in time at all interior points. For the predictor step of the calculations, the averaged coefficient values designated by double subscripts are replaced by the corresponding values at the current time level solution surface.

B.2.3.2 BLADE SURFACE BOUNDARY POINT UNIT PROCESS. In a two-dimensional cascade flow, the surface of the blade forms the only solid wall boundary. The boundary condition applicable at a free slip boundary point is that the velocity normal to the boundary is zero.

$$\hat{n}_1 \cdot \vec{V}_6 = 0 \quad (\text{B.37})$$

where \hat{n}_1 is the unit vector normal to the blade surface and pointing into the blade at the solution point under consideration.

Since the \hat{n}_1 unit vector lies outside the flowfield, the waveline equation corresponding to the \hat{n}_1 unit vector is replaced by the boundary condition, equation (B.37). Since one of the four waveline equations used at interior points is not available, Butler's technique must be applied in a different manner at blade surface boundary points.

Expanding equation (B.37) to scalar form and solving for the v velocity component at the solution point yields:

$$v_6 = \left(\frac{n_{2j}}{n_{2i}} \right) u_6 \quad (\text{B.38})$$

where the unit vector \hat{n}_2 is tangent to the blade surface at the solution point.

Subtracting the \hat{n}_2 waveline equation, equation (B.27), from the \hat{n}_4 waveline equation, equation (B.29), using equation (B.38) to eliminate the v velocity component from the result, and rearranging yields:

$$u_6 = \frac{n_{2i} [C_4 - C_2]}{[(\rho a)_{64} + (\rho a)_{62}]} \quad (B.39)$$

Subtracting the noncharacteristic relation from the sum of the \hat{n}_3 waveline equation plus the \hat{n}_2 and \hat{n}_4 waveline equations added and divided by two, and implementing the boundary condition yields:

$$P_6 = \frac{1}{2} \left\{ (n_{2i} u_6 + n_{2j} v_6) [(\rho a)_{62} - (\rho a)_{64}] + C_2 + C_4 \right\} + C_3 - C_5 \quad (B.40)$$

The chosen boundary condition reveals that the pathline lies on the blade surface. Thus the pathline equation, equation (B.6), is still applicable and is repeated here in finite difference form:

$$\rho_6 = \rho_5 + \frac{1}{a_{56}^2} [P_6 - P_5] \quad (B.41)$$

Equations (B.38) to (B.41) are solved to advance the solution in time at all solid wall boundary points. For the predictor step of the calculations, the averaged coefficient values designated by double subscripts are replaced by the corresponding values at the current time level solution surface.

B.2.3.3 TRAILING EDGE POINT UNIT PROCESS. In the inviscid cascade flowfield, the Kutta condition is enforced at the trailing edge of the cascade blade. Specifically, the flows leaving the two sides of the blade surface at the trailing edge are forced to flow parallel to each other in a direction such that the static pressure is equal on the two sides of the trailing edge point.

In the present investigation, the requirements listed above are enforced by installing a small imaginary solid wall segment which is hinged at the trailing edge point. By forcing the flow on both sides of the wall segment to follow the wall, the flows on the two sides of the trailing edge are forced to be parallel to each other. The flow on each side of the hinged wall segment is solved independently, in the same manner as described in Section B.2.3.2 for the solid blade surface point. Therefore, as at the solid wall points, equations (B.38) to (B.41) are solved to determine the forward time level property values.

After solving for the flow properties on each side of the hinged wall segment independently, the pressure difference between the two sides is compared. If the difference is not equal to zero, to within a specified tolerance, a zero finding secant method is used to correct the wall orientation (flow angle) toward the angle where the pressures are equal. This procedure is iterated until the pressure difference between the pressure and suction sides of the blade, at the trailing edge, is approximately zero.

Because the imaginary hinged wall segment is not stationary, but instead is allowed to pivot, the local coordinate systems, set up on each side of the wall at the trailing edge points, also pivot. During the solution process at the trailing edge point, the local coordinate system unit vectors, \hat{n}_1 , \hat{n}_2 , \hat{n}_3 , and \hat{n}_4 , are defined with respect to their orientation at the forward time level, that is, the time level of the predicted and corrected properties. In other words, at each iteration, \hat{n}_1 and \hat{n}_3 are aligned perpendicular to, and \hat{n}_2 and \hat{n}_4 are aligned parallel to the hinged wall segment at the angle the wall is assumed to be at, at the forward time level, during that iteration.

B.2.3.4 EXIT BOUNDARY POINT UNIT PROCESSES. Because the component of the flow velocity which crosses the exit boundary can be either subsonic or supersonic, each exit boundary point must be checked, at each time step, to determine which unit process is appropriate. At the exit boundary, the unit vector \hat{n}_1 is placed perpendicular to the boundary, pointing outward. Therefore, the quantity $\hat{n}_1 \cdot \vec{V}$ is calculated and compared to the local speed of sound, a , to determine whether the boundary point should be handled as a supersonic or a subsonic exit boundary point.

When the component of the fluid velocity perpendicular to the exit boundary is greater than or equal to the local speed of sound, the exit flow is only dependent upon upstream conditions. From a characteristic perspective, this means that the base of the Mach cone (domain of dependence) lies entirely within the computational domain. In this case the interior point unit process is appropriate and is applied as described in Section B.2.3.1.

When the component of the fluid velocity perpendicular to the exit boundary is less than the local speed of sound, the fluid properties at that boundary are dependent upon both upstream and downstream phenomena. Therefore, at least one boundary condition must be applied at the boundary point. The subsonic exit point unit process is described next.

In the characteristics analysis developed for the present investigation, one boundary condition is applied at a subsonic exit boundary point. The chosen boundary condition is that the exit static pressure is known:

$$P = P_{\text{exit}} \quad (\text{B.42})$$

The waveline equation corresponding to the \hat{n}_1 unit normal vector is outside the flowfield, and is replaced by the boundary condition, equation (B.42). Therefore, the equations used in conjunction with the Butler method to determine the applicable equations at a subsonic exit boundary are the boundary condition, equation (B.42), the finite difference form of the noncharacteristic relation, equation (B.26) and the three remaining waveline equations corresponding to the unit normal vectors \hat{n}_2 , \hat{n}_3 , and \hat{n}_4 in finite difference form, equations (B.28) to (B.30).

The finite difference form of the pathline equation, equation (B.25), is used to solve for the density at the subsonic exit boundary points. That equation is repeated here for convenience:

$$\rho_6 = \rho_5 + \frac{1}{a_{56}^2} [P_6 - P_5] \quad (B.43)$$

Subtracting the \hat{n}_2 waveline equation, equation (B.28) from the \hat{n}_4 waveline equation, equation (B.30), yields:

$$(n_{2i} u_6 + n_{2j} v_6) \left[(\rho a)_{64} + (\rho a)_{62} \right] = [C_4 - C_2] \quad (B.44)$$

Subtracting the noncharacteristic relation from the sum of the \hat{n}_3 waveline equation plus the \hat{n}_2 and \hat{n}_4 waveline equations added and divided by two yields:

$$\begin{aligned} P_6 + (\rho a)_{63} (n_{1i} u_6 + n_{1j} v_6) + \frac{1}{2} (n_{2i} u_6 + n_{2j} v_6) \left[(\rho a)_{64} - (\rho a)_{62} \right] \\ = \frac{1}{2} (C_2 + C_4) + C_3 - C_5 \end{aligned} \quad (B.45)$$

Solving equations (B.44) and (B.45) simultaneously for the two components of velocity and implementing the boundary condition produces:

$$v_6 = \frac{\left[\frac{C_4 - C_2}{(\rho a)_{64} + (\rho a)_{62}} \right] - n_{2i} u_6}{n_{2j}} \quad (B.46)$$

$$u_6 = \left(\frac{n_{1i}^2}{n_{1j}} \right) \left(\frac{C_4 - C_2}{(\rho a)_{64} + (\rho a)_{62}} \right) + \frac{n_{1i}}{(\rho_{63})} \left[\frac{1}{2} \left\{ \left(\frac{(\rho a)_{64} - (\rho a)_{62}}{(\rho a)_{64} + (\rho a)_{62}} \right) (C_2 - C_4) + C_2 + C_4 \right\} + C_3 - C_5 - P_{\text{exit}} \right] \quad (B.47)$$

At a subsonic exit point, the boundary condition defines the exit pressure at the solution point, the pathline equation in finite difference form, equation (B.43), is solved for the density, and equations (B.46) and (B.47) are solved for the velocity components to advance the solution in time. For the predictor step of the calculations, the averaged coefficient values designated by double subscripts are replaced by the corresponding values at the current time level solution surface.

B.2.3.5 SUBSONIC INFLOW BOUNDARY POINT UNIT PROCESS. As in the MacCormack code, the stagnation pressure, P_0 , the stagnation temperature, T_0 , and the flow angle, θ , are specified as boundary conditions at the inflow boundary. Of the compatibility equations, only the waveline corresponding to the \hat{n}_1 unit vector, equation (B.27), is solved at the inflow boundary. The \hat{n}_1 unit vector is chosen to be the x-direction unit vector, \hat{i} . Therefore, the equations applicable at the subsonic inflow boundary are the

boundary conditions and one waveline equation:

$$P_0 = P_{0_{\text{inflow}}} \quad (\text{B.48})$$

$$T_0 = T_{0_{\text{inflow}}} \quad (\text{B.49})$$

$$\theta = \theta_{\text{inflow}} \quad (\text{B.50})$$

$$P_6 - (\rho a)_{61} u_6 + \frac{\Delta t}{2} (\rho a^2 v_y)_6 = C_1 \quad (\text{B.51})$$

Since only one compatibility equation is used at the subsonic inflow boundary, it must be computed without simplification. This is achieved through iteration.

The Mach number, M , is the iteration variable in the subsonic inflow boundary point unit process. Knowing an estimated forward time Mach number and the boundary conditions given by equations (B.48) to (B.50), several additional properties can be computed. In terms of known properties of the flow and the Mach number:

$$\rho = \frac{P_0}{RT_0} \left(\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right)^{\frac{1}{\gamma-1}} \quad (\text{B.52})$$

$$P = P_0 \left(\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right)^{\frac{\gamma}{\gamma-1}} \quad (\text{B.53})$$

$$u = \frac{M (\gamma R T_0)^{\frac{1}{2}}}{\left(1 + \tan^2(\theta)\right)^{\frac{1}{2}} \left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{1}{2}}} \quad (\text{B.54})$$

$$v = u \tan(\theta) \quad (\text{B.55})$$

The iterative process used to find property values at subsonic inflow boundary points is described next. For both predictor and corrector steps, the procedure is initiated by guessing the forward time Mach number at the inflow point being considered. For the first time step, the Mach number guess is calculated from the property values at the initial-value surface. Subsequently the initial Mach number guess is calculated from the property values at the current time level. The guessed Mach number is used in equations (B.52) to (B.55) to calculate guessed forward time values of ρ , u , v , and P .

Equation (B.51) is solved for the pressure at the solution point:

$$P_6 = (\rho a)_{61} u_6 - \frac{\Delta t}{2} (\rho a^2 v_y)_6 + C_1 \quad (\text{B.56})$$

In equation (B.56) the value of v_y at the forward time level solution point is approximated by the value of v_y at the current time level solution point. This lagging of the cross derivative reduces the method to first-order in time at the inflow boundary points. As the solution is marched in time to a steady solution, the first-order error term will approach zero. This handling of the cross derivative term is necessary to avoid violating the physical domain of dependence at the forward time level.

Equations (B.52) to (B.55) are used to calculate guessed values of the solution properties at the forward time level. These guessed property values are used in equation (B.56) to calculate the pressure at the forward time solution point. If the calculated value of pressure and the guessed value of pressure are equal, to within a specified tolerance, the Mach number guess is correct and the guessed property values are the forward time level values. If the calculated and guessed values are not approximately equal, a zero finding secant method is used to home in on the correct Mach number. For the results presented in Section VII, a tolerance of 10^{-6} was used to evaluate convergence of the pressure difference.

For the predictor step of the calculations, the averaged coefficient values designated by double subscripts are replaced by the corresponding values at the current time level solution surface.

B.2.4 SOLUTION PROCEDURE. At each time level all the unit processes discussed in Appendix B.2.3 are applied to a given cascade geometry and a current time level solution surface to advance the solution in time.

The flowfield initial conditions are set in the same manner used by the MacCormack flow solver except that primitive variables are set rather than conservation variables. The MacCormack code initial-value surface is described in Section VI.1.

The time step calculation is based on the Courant Friedrichs Lewy (CFL) [24] stability criterion. A procedure similar to the one described in Section VI.2.1 is used to conservatively approximate the CFL time limit at each grid

point. The characteristics code and MacCormack code time step limit analyses differ only in so far as the applicable convex hulls illustrated in Figures B-3 and 6-1 differ.

The MacCormack code requires explicit smoothing to converge. Since the characteristics code does not require smoothing for the cases discussed in Section VII, none is applied. The capability of the code could be extended well into the transonic range with the addition of explicit smoothing.

In the present investigation the characteristics code is marched in time until a steady solution is achieved. The convergence check, used at each time step, consists of comparing the largest normalized change in pressure at any solution point in the flowfield to a specified tolerance. The current time level pressure at each point is used to normalize the change in pressure at that point. For the results discussed in Section VII, solutions were converged to a tolerance of 10^{-5} .

APPENDIX C

CONSERVATION VARIABLE COMPATIBILITY EQUATIONS

This Appendix presents the primitive variable and conservative variable compatibility equations and demonstrates the equivalence of these two sets of equations.

C.1 COMPATIBILITY EQUATIONS

As described in Appendix B, characteristic theory uses the primitive variable form of the governing differential equations of motion to derive the pathline and waveline equations. The primitive variable compatibility equations are repeated here for convenience.

The pathline equation is the energy equation in primitive variable form.

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (C.1)$$

The waveline equation is formed by the following linear combination of the primitive variable governing equations:

$$a^2 \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{continuity} \\ \text{equation} \end{bmatrix} - \rho a \hat{n} \cdot \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{vector} \\ \text{momentum} \\ \text{equation} \end{bmatrix} + \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{energy} \\ \text{equation} \end{bmatrix} = 0 \quad (C.2)$$

which reduces to:

$$\frac{\partial P}{\partial t} - \rho a \hat{n} \cdot \frac{\partial \vec{V}}{\partial t} + \rho a^2 \left[\nabla \cdot \vec{V} - \hat{n} \cdot (\hat{n} \cdot \nabla) \vec{V} \right] = 0 \quad (C.3)$$

Because these equations are cast in primitive variables, the Kentzer method [2] is normally applied in primitive variables. However, a conservation variable variation of the Kentzer method is being developed and applied in the present investigation.

The conservation variable form of the governing equations written in vector notation are repeated here for convenience:

$$(\rho)_t = \mathcal{C} \quad (C.4)$$

$$(\rho \vec{V})_t = \vec{\mathcal{M}} \quad (C.5)$$

$$(\rho e)_t = \mathcal{E} \quad (C.6)$$

where

$$\mathcal{C} = -\nabla \cdot (\rho \vec{V}) \quad (C.7)$$

$$\vec{\mathcal{M}} = -\nabla \cdot (\rho \vec{V} \vec{V}) - \nabla P \quad (C.8)$$

$$\mathcal{E} = -\nabla \cdot [\vec{V}(\rho e + P)] \quad (C.9)$$

The following linear combination of the conservation variable form of the governing equations produces the conservation variable pathline equation:

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma - 1} \right] \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{continuity} \\ \text{equation} \end{bmatrix} - \vec{V} \cdot \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{vector} \\ \text{momentum} \\ \text{equation} \end{bmatrix} + \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{energy} \\ \text{equation} \end{bmatrix} = 0 \quad (C.10)$$

Substituting equations (C.4) to (C.6) into equation (C.10) gives the conservation variable form of the pathline equation in vector notation.

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \bar{\nabla} \cdot (\rho \bar{\nabla})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \bar{\nabla} \cdot \bar{\mathcal{M}} + \mathcal{E} \quad (C.11)$$

In a similar manner, the following linear combination of the conservation variable form of the governing equations produces the conservation variable waveline equation:

$$\left[\frac{V^2}{2} + \frac{a \hat{n} \cdot \bar{\nabla}}{\gamma-1} \right] \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{continuity} \\ \text{equation} \end{bmatrix} - \left[\bar{\nabla} + \frac{a \hat{n}}{\gamma-1} \right] \cdot \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{vector} \\ \text{momentum} \\ \text{equation} \end{bmatrix} + \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{energy} \\ \text{equation} \end{bmatrix} = 0 \quad (C.12)$$

Substituting equations (C.4) to (C.6) into equation (C.12) gives the conservation variable waveline equation in vector notation.

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n} \cdot \bar{\nabla}}{\gamma-1} \right] \rho_t - \left[\bar{\nabla} + \frac{a \hat{n}}{\gamma-1} \right] \cdot (\rho \bar{\nabla})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n} \cdot \bar{\nabla}}{\gamma-1} \right] \mathcal{E} - \left[\bar{\nabla} + \frac{a \hat{n}}{\gamma-1} \right] \cdot \bar{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (C.13)$$

By applying the Kentzer method to the compatibility equations, equations (C.11) and (C.13), the appropriate set of conservation variable equations is derived for each type of boundary condition. These conservative variable formulations are used in the present investigation.

It is important to note that the conservation variable compatibility equations have not been cast in terms of directional derivatives along characteristic surfaces. Therefore, these conservation variable compatibility equations can not be used to develop a numerical method of characteristics based on conservation variables.

C.2 EQUIVALENCE OF CONSERVATION VARIABLE AND PRIMITIVE VARIABLE COMPATIBILITY EQUATIONS

Equations (C.4) to (C.6) can be expanded into a variety of primitive variable forms. The following primitive variable continuity, vector momentum, and energy equations have been shown, in Appendix A, to be equivalent to equations (C.4) to (C.6).

For the continuity equation:

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) = 0 \quad (C.14)$$

For the vector momentum equation:

$$\vec{V} \frac{D\rho}{Dt} + \rho \frac{D\vec{V}}{Dt} + \rho \vec{V}(\nabla \cdot \vec{V}) + \nabla P = 0 \quad (C.15)$$

or

$$\rho \frac{D\vec{V}}{Dt} + \nabla P = 0 \quad (C.16)$$

For the energy equation:

$$\frac{1}{\gamma-1} \frac{DP}{Dt} + \frac{V^2}{2} \frac{D\rho}{Dt} + \rho \frac{D\frac{V^2}{2}}{Dt} + \frac{\gamma P}{\gamma-1} (\nabla \cdot \vec{V}) + \rho \frac{V^2}{2} (\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P = 0 \quad (C.17)$$

or

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (C.18)$$

C.2.1 PATHLINE COMPATIBILITY EQUATION. The equivalence of the conservation variable pathline equation and the primitive variable pathline equation is demonstrated in this section.

Substituting the governing equations, which have been expanded into primitive variables, equations (C.14), (C.15), and (C.17), into equation (C.10) yields:

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \left[\frac{D\rho}{Dt} + \rho (\nabla \cdot \vec{V}) \right] - \vec{V} \cdot \left[\vec{V} \left(\frac{D\rho}{Dt} + \rho (\nabla \cdot \vec{V}) \right) + \rho \frac{D\vec{V}}{Dt} + \nabla P \right] + \frac{1}{\gamma-1} \frac{DP}{Dt} + \frac{V^2}{2} \frac{D\rho}{Dt} + \rho \frac{D\frac{V^2}{2}}{Dt} + \frac{\gamma P}{\gamma-1} (\nabla \cdot \vec{V}) + \rho \frac{V^2}{2} (\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P = 0 \quad (C.19)$$

Distributing the coefficients of the continuity and momentum equation terms yields:

$$\begin{aligned}
& \left[\frac{V^2}{2} \right] \frac{D\rho}{Dt} + \left[\frac{V^2}{2} \right] \rho (\nabla \cdot \vec{V}) - \left[\frac{a^2}{\gamma-1} \right] \frac{D\rho}{Dt} - \left[\frac{a^2}{\gamma-1} \right] \rho (\nabla \cdot \vec{V}) \\
& - V^2 \left[\frac{D\rho}{Dt} \right] - \rho V^2 (\nabla \cdot \vec{V}) - \rho \vec{V} \cdot \frac{D\vec{V}}{Dt} - \vec{V} \cdot \nabla P + \frac{1}{\gamma-1} \frac{DP}{Dt} \\
& + \frac{V^2}{2} \frac{D\rho}{Dt} + \rho \frac{D \frac{V^2}{2}}{Dt} + \frac{\gamma P}{\gamma-1} (\nabla \cdot \vec{V}) + \rho \frac{V^2}{2} (\nabla \cdot \vec{V}) + \vec{V} \cdot \nabla P = 0 \quad (C.20)
\end{aligned}$$

Canceling opposite terms and grouping like terms leaves:

$$\frac{1}{\gamma-1} \frac{DP}{Dt} - \frac{a^2}{\gamma-1} \frac{D\rho}{Dt} + \left[\frac{\gamma P}{\gamma-1} - \frac{a^2 \rho}{\gamma-1} \right] (\nabla \cdot \vec{V}) + \rho \left[-\vec{V} \cdot \frac{D\vec{V}}{Dt} + \frac{D \frac{V^2}{2}}{Dt} \right] = 0 \quad (C.21)$$

Since $\vec{V} \cdot \frac{D\vec{V}}{Dt} = \frac{D(V^2/2)}{Dt}$, the last term in equation (C.21) is zero. In addition,

$a^2 = \gamma P / \rho$, for a perfect gas, so the third term in equation (C.21) is also zero.

Therefore, equation (C.21) reduces to:

$$\frac{DP}{Dt} - a^2 \frac{D\rho}{Dt} = 0 \quad (C.22)$$

Thus, the conservation variable form of the pathline equation is equivalent to the primitive variable form of the pathline equation.

C.2.2 WAVELINE COMPATIBILITY EQUATION. The equivalence of the conservation variable waveline equation and the primitive variable waveline equation is demonstrated in this section.

In Section C.2.1 the conservation variable pathline equation, equation (C.10), is shown to be equivalent to the primitive variable pathline equation, equation (C.1). Therefore, if the conservation variable pathline equation, equation (C.10), is subtracted from the conservation variable waveline equation, equation (C.12), and the primitive variable pathline equation, equation (C.1), is subtracted from the primitive variable waveline equation, equation (C.2), the resulting two equations will be equivalent if the conservation variable and primitive variable forms of the waveline compatibility equation are equivalent.

Subtracting equation (C.10) from equation (C.12) leaves:

$$\left[\frac{a\hat{n} \cdot \vec{V} + a^2}{\gamma - 1} \right] \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{continuity} \\ \text{equation} \end{bmatrix} - \left[\frac{a\hat{n}}{\gamma - 1} \right] \cdot \begin{bmatrix} \text{conservation} \\ \text{variable} \\ \text{vector} \\ \text{momentum} \\ \text{equation} \end{bmatrix} = 0 \quad (\text{C.23})$$

Subtracting equation (C.1) from equation (C.2) leaves:

$$a^2 \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{continuity} \\ \text{equation} \end{bmatrix} - \rho a \hat{n} \cdot \begin{bmatrix} \text{primitive} \\ \text{variable} \\ \text{vector} \\ \text{momentum} \\ \text{equation} \end{bmatrix} = 0 \quad (\text{C.24})$$

Inserting equations (C.14) and (C.16) into equation (C.24) and rearranging to form substantial derivatives where possible yields:

$$a^2 \left[\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) \right] - \rho a \hat{n} \cdot \left[\frac{D\vec{V}}{Dt} + \frac{\nabla P}{\rho} \right] = 0 \quad (\text{C.25})$$

Substituting equations (C.14) and (C.15) into equation (C.23) yields:

$$\left[\frac{a\hat{n} \cdot \vec{V} + a^2}{\gamma - 1} \right] \left[\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) \right] - \frac{a\hat{n}}{\gamma - 1} \cdot \left[\vec{V} \left[\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) \right] + \rho \frac{D\vec{V}}{Dt} + \nabla P \right] = 0 \quad (C.26)$$

Rearranging equation (C.26) yields:

$$\begin{aligned} (a\hat{n} \cdot \vec{V}) \frac{D\rho}{Dt} + (a\hat{n} \cdot \vec{V}) \rho(\nabla \cdot \vec{V}) + a^2 \left[\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) \right] - (a\hat{n} \cdot \vec{V}) \frac{D\rho}{Dt} \\ - (a\hat{n} \cdot \vec{V}) \rho(\nabla \cdot \vec{V}) - \rho a\hat{n} \cdot \left[\frac{D\vec{V}}{Dt} \right] - \rho a\hat{n} \cdot \frac{\nabla P}{\rho} = 0 \end{aligned} \quad (C.27)$$

Canceling opposite terms leaves:

$$a^2 \left[\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) \right] - \rho a\hat{n} \cdot \left[\frac{D\vec{V}}{Dt} + \frac{\nabla P}{\rho} \right] = 0 \quad (C.28)$$

Equation (C.28) is identical to equation (C.25). Therefore, the conservation variable waveline equation is equivalent to the primitive variable waveline equation.

APPENDIX D

UNIT PROCESSES

Several unit processes are employed in the computation of cascade flowfields. Each of the following types of flowfield points must be considered:

1. interior points,
2. solid wall boundary points,
3. trailing edge point,
4. subsonic inlet boundary points,
5. subsonic exit boundary points, and
6. supersonic exit boundary points.

All periodic boundary points and grid cut points lie in one of the regions listed above. Therefore, no additional sets of equations are required to compute these points. However, because of their location on the computational grid, they do require special handling of indices.

Appendix D.1 summarizes the MacCormack method [3] and the Kentzer method [2] prior to deriving the unit processes listed above.

The equations derived in this section are presented in terms of physical domain (x,y,t) derivatives. Prior to their implementation, these equations are transformed into computational space. At interior points the governing equations are transformed into the strong conservation-law form as shown in

equation (2.33). At the boundaries the transformed equations consists of the same groupings of space derivatives transformed into strong conservation-law form. However, these derivative terms have coefficients which are not affected by the transformation. Therefore, the boundary point equations are in a weak conservation form.

At interior points the Euler equations apply. Those equations are repeated here for convenience:

$$(\rho)_t = \mathcal{C} \quad (\text{D.1})$$

$$(\rho \vec{V})_t = \vec{\mathcal{M}} \quad (\text{D.2})$$

$$(\rho e)_t = \mathcal{E} \quad (\text{D.3})$$

where \mathcal{C} , $\vec{\mathcal{M}}$, and \mathcal{E} contain the space derivatives appearing in the continuity, vector momentum, and energy equations, respectively. Written in vector notation, \mathcal{C} , $\vec{\mathcal{M}}$, and \mathcal{E} are:

$$\mathcal{C} = -\nabla \cdot (\rho \vec{V}) \quad (\text{D.4})$$

$$\vec{\mathcal{M}} = -\nabla \cdot (\rho \vec{V} \vec{V}) - \nabla P \quad (\text{D.5})$$

$$\mathcal{E} = -\nabla \cdot [\vec{V}(\rho e + P)] \quad (\text{D.6})$$

At all boundary points, the Kentzer method is used to determine the appropriate set of equations to be solved. At all interior and all boundary points, the space derivative terms \mathcal{C} , $\vec{\mathcal{M}}$, and \mathcal{E} , appear. In addition, at all interior and all boundary points, the MacCormack explicit finite difference

method is used to discretize the appropriate equations.

D.1 THE MACCORMACK METHOD

The MacCormack explicit finite difference method [2] is used in this investigation to calculate the flowfield solution. This predictor-corrector method is second-order accurate in time and space.

The transformed governing equations in matrix form are presented in Section II.3.3 and are repeated here for convenience:

$$\bar{Q}_t + \bar{E}_\xi + \bar{F}_\eta = 0 \quad (D.7)$$

where

$$\bar{Q} = I \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad \bar{E} = \begin{bmatrix} I(\rho U) \\ I(\rho u U + \xi_x P) \\ I(\rho v U + \xi_y P) \\ I[(\rho e + P)U] \end{bmatrix} \quad \bar{F} = \begin{bmatrix} I(\rho V) \\ I(\rho u V + \eta_x P) \\ I(\rho v V + \eta_y P) \\ I[(\rho e + P)V] \end{bmatrix}$$

The two steps of the MacCormack method can be summarized, for the present investigation, by the following equations. The predictor space derivatives are all computed using current time level property values:

$$\bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left[\mathbf{F} \left(\bar{E}_\xi \right)_{i,j}^n + \mathbf{F} \left(\bar{F}_\eta \right)_{i,j}^n \right] \quad (D.8)$$

The corrector space derivatives are all computed using forward time level property values:

$$\bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left[\mathbf{B} \left(\bar{E}_\xi \right)_{i,j}^{n+1} + \mathbf{B} \left(\bar{F}_\eta \right)_{i,j}^{n+1} \right] \quad (D.9)$$

To achieve second order accuracy in time and space, the results of these two

steps are averaged:

$$\bar{Q}_{i,j}^{n+1} = \frac{1}{2} \left[\bar{Q}_{i,j}^{n+1} + \bar{Q}_{i,j}^{n+1} \right] \quad (D.10)$$

Efficiency is improved by combining equations (D.9) and (D.10) in the following manner:

$$\bar{Q}_{i,j}^{n+1} = \frac{1}{2} \left[\bar{Q}_{i,j}^{n+1} + \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left\{ \mathbf{B} \left(\bar{E}_\xi \right)_{i,j}^{n+1} + \mathbf{B} \left(\bar{F}_\eta \right)_{i,j}^{n+1} \right\} \right] \quad (D.11)$$

As shown in equations (D.8), (D.9), and (D.11), $\mathbf{F}()$ are first-order forward-difference operators and $\mathbf{B}()$ are first-order backward-difference operators. Those operators are:

$$\mathbf{F}(\bar{E}_\xi^n) = \frac{(\bar{E}_{i+1,j}^n - \bar{E}_{i,j}^n)}{\Delta \xi}$$

$$\mathbf{F}(\bar{F}_\eta^n) = \frac{(\bar{F}_{i,j+1}^n - \bar{F}_{i,j}^n)}{\Delta \eta}$$

$$\mathbf{B}(\bar{E}_\xi^{n+1}) = \frac{(\bar{E}_{i,j}^{n+1} - \bar{E}_{i-1,j}^{n+1})}{\Delta \xi}$$

$$\mathbf{B}(\bar{F}_\eta^{n+1}) = \frac{(\bar{F}_{i,j}^{n+1} - \bar{F}_{i,j-1}^{n+1})}{\Delta \eta}$$

By incorporating the forward-difference operators, the MacCormack predictor equation, equation (D.8), applied to the transformed governing equations in matrix form becomes:

$$\bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n - \Delta t J_{i,j} \left[\frac{\bar{E}_{i+1,j}^n - \bar{E}_{i,j}^n}{\Delta \xi} + \frac{\bar{F}_{i,j+1}^n - \bar{F}_{i,j}^n}{\Delta \eta} \right] \quad (D.12)$$

Similarly, by incorporating the backward-difference operators, the MacCormack corrector equation, equation (D.11), applied to the transformed governing equations in matrix form becomes:

$$\bar{Q}_{i,j}^{n+1} = 0.5 \left[\bar{Q}_{i,j}^n + \bar{Q}_{i,j}^{n+1} - \Delta t J_{i,j} \left\{ \frac{\bar{E}_{i,j}^{n+1} - \bar{E}_{i-1,j}^{n+1}}{\Delta \xi} + \frac{\bar{F}_{i,j}^{n+1} - \bar{F}_{i,j-1}^{n+1}}{\Delta \eta} \right\} \right] \quad (D.13)$$

For the present investigation, all grid spacings on the computational grid are unity (i.e., $\Delta \xi = \Delta \eta = 1.0$).

At each of the boundaries, one of the forward- or the backward-difference operations requires flow properties and transformation metrics outside the flowfield to approximate flow property partial derivatives at the boundaries. Therefore, first-order or second-order extrapolations are used to approximate these values outside boundaries. Then the MacCormack method, as described above, is applied to the appropriate boundary point equations to evaluate boundary property values.

D.2 THE KENTZER METHOD

As described in Section III, the Kentzer method is being applied, in the present investigation, in conservation variables. Therefore, at each of the boundaries, the conservation variable pathline equation, equation (3.12), and the conservation variable waveline equation, equation (3.13), combine with the required boundary conditions to form the appropriate set of equations to describe the pertinent physics. Equations (3.12) and (3.13) are repeated here for convenience:

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{\nabla} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{\nabla} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (D.14)$$

$$\begin{aligned} \left[\frac{V^2}{2} + \frac{a \hat{n} \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{\nabla} + \frac{a \hat{n}}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ = \left[\frac{V^2}{2} + \frac{a \hat{n} \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{\nabla} + \frac{a \hat{n}}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (D.15)$$

In order to implement the boundary point equations, a local coordinate system is established at the boundary point. The local boundary coordinates are defined by \hat{b} and \hat{c} , where $\hat{b} = b_i \hat{i} + b_j \hat{j}$ is the unit vector normal to the boundary and $\hat{c} = c_i \hat{i} + c_j \hat{j}$ is the unit vector tangent to the boundary.

At any given point, an infinite number of wave surfaces exist which could be chosen for application of the waveline equation. Since a maximum of four equations can be independent in an unsteady two-dimensional flow, three waveline equations are used in conjunction with the pathline equation to describe the properties at a point. The wave surface unit normal vectors, \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , are chosen such that they are equally distributed and aligned with the boundary under consideration. The \hat{n}_1 wave surface unit normal vector is chosen to be the unit vector normal to the boundary, \hat{b} . Thus, the unit normal vectors are defined in the following manner:

$$\hat{n}_1 = \hat{b} \quad (D.16)$$

$$\hat{n}_2 = -\frac{1}{2} \hat{b} + \frac{\sqrt{3}}{2} \hat{c} \quad (D.17)$$

$$\hat{n}_3 = -\frac{1}{2} \hat{b} - \frac{\sqrt{3}}{2} \hat{c} \quad (D.18)$$

All five of these unit vectors, \hat{b} , \hat{c} , \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , are shown at three boundary point locations on a C-grid in Figure D-1.

D.3 INTERIOR POINT UNIT PROCESS

At interior points, the MacCormack method is applied directly to the transformed governing equations in the strong conservation-law form, equation (2.33). The resulting finite difference equations are used to advance the solution in time.

D.4 BLADE SURFACE BOUNDARY POINT UNIT PROCESS

In a two-dimensional cascade flow, the surface of the blade forms the only solid wall boundary. The boundary condition applicable at a free slip solid boundary point is that the velocity normal to the boundary is zero:

$$\hat{b} \cdot \vec{V} = 0 \quad (D.19)$$

Multiplication by the density yields an equivalent boundary condition:

$$\hat{b} \cdot (\rho \vec{V}) = 0 \quad (D.20)$$

Since the blade geometry and the boundary condition are invariant with respect to time, the following is also a valid boundary condition:

$$\hat{b} \cdot (\rho \vec{V})_t = 0 \quad (D.21)$$

Expanding equation (D.20) to scalar form and rearranging provides an expression for the y-momentum, ρv , in terms of the x-momentum, ρu :

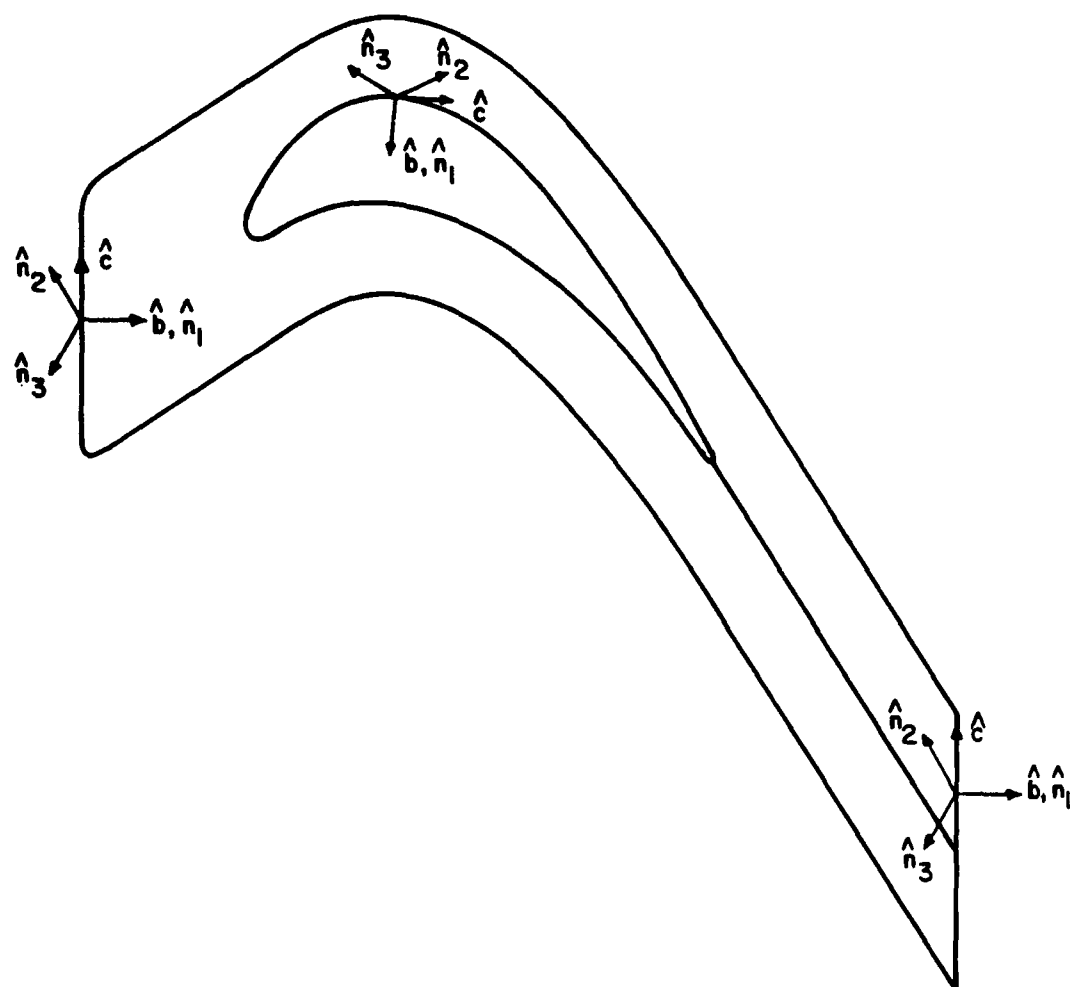


Figure D-1. Unit vectors \hat{b} , \hat{c} , \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 on three boundaries.

$$\rho v = \left[-\frac{b_i}{b_j} \right] \rho u = \left[\frac{c_j}{c_i} \right] \rho u \quad (\text{D.22})$$

To determine the applicable equations at the blade surface boundary point, the governing equations, equations (D.1) to (D.3), are replaced by compatibility equations, equations (D.14) and (D.15). The waveline equation, equation (D.15), is applied in three wave surfaces corresponding to the wave surface unit normal vectors \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (D.16) to (D.18). Since the boundary condition, equation (D.19), must be incorporated, the waveline equation corresponding to vector \hat{n}_1 (which was intentionally placed outside the flowfield) is replaced by the boundary condition. The applicable equations at a solid wall boundary point are the boundary condition, equation (D.19), the pathline equation, equation (D.14), and the waveline equation, equation (D.15), applied in the remaining two wave surfaces corresponding to wave surface unit normal vectors \hat{n}_2 and \hat{n}_3 . Thus,

$$\hat{b} \cdot \vec{V} = 0 \quad (\text{D.23})$$

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{V} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{V} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (\text{D.24})$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ & = \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (\text{D.25})$$

$$\begin{aligned}
& \left[\frac{V^2}{2} + \frac{a\hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a\hat{n}_3}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\
& = \left[\frac{V^2}{2} + \frac{a\hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a\hat{n}_3}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (D.26)
\end{aligned}$$

Equations (D.23) to (D.26) comprise an appropriate set of equations for the solid wall boundary point. However, these equations are unnecessarily complicated. Equations (D.24) to (D.26) each contain more than one time derivative. These four equations can be rearranged into a form which is simpler and more computationally efficient to solve by a finite difference method. The rearrangement and simplification of this set of equations is demonstrated next.

Adding the two waveline equations, equations (D.25) and (D.26), and rearranging yields:

$$\begin{aligned}
& \left[V^2 + \frac{a(\hat{n}_2 + \hat{n}_3) \cdot \vec{V}}{\gamma-1} \right] (\rho_t - \mathcal{E}) \\
& - \left[2\vec{V} + \frac{a(\hat{n}_2 + \hat{n}_3)}{\gamma-1} \right] \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] + 2[(\rho e)_t - \mathcal{E}] = 0 \quad (D.27)
\end{aligned}$$

Subtracting twice the pathline equation, equation (D.24), from equation (D.27), using the result that $\hat{n}_2 + \hat{n}_3 = -\hat{b}$, and rearranging yields:

$$\left[\hat{b} \cdot \vec{V} - 2a \right] (\rho_t - \mathcal{E}) = \hat{b} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (D.28)$$

Implementing the boundary condition and solving for the temporal derivative of density leaves:

$$\rho_t = \mathcal{C} + \left[\frac{1}{2a} \right] \hat{b} \cdot \vec{\mathcal{M}} \quad (\text{D.29})$$

Equation (D.29) is solved for the density at the solid blade surface boundary point.

The derivation of the next solid wall boundary point equation starts by finding another expression for the term $(\rho_t - \mathcal{C})$. Subtracting the \hat{n}_3 waveline equation, equation (D.26), from the \hat{n}_2 waveline equation, equation (D.25), recognizing that $\hat{n}_2 - \hat{n}_3 = \sqrt{3} \hat{c}$, and simplifying the result produces:

$$(\hat{c} \cdot \vec{\nabla})(\rho_t - \mathcal{C}) = \hat{c} \cdot [(\rho \vec{\nabla})_t - \vec{\mathcal{M}}] \quad (\text{D.30})$$

Adding \hat{b} times equation (D.28) to \hat{c} times equation (D.30) yields:

$$(\rho \vec{\nabla})_t - \vec{\mathcal{M}} = \left[\hat{b}(\hat{b} \cdot \vec{\nabla} - 2a) + \hat{c}(\hat{c} \cdot \vec{\nabla}) \right] (\rho_t - \mathcal{C}) \quad (\text{D.31})$$

Simplifying equation (D.31) leaves:

$$(\rho \vec{\nabla})_t = \vec{\mathcal{M}} + \left[(\vec{\nabla} - 2a\hat{b}) \right] (\rho_t - \mathcal{C}) \quad (\text{D.32})$$

Using equation (D.28) to eliminate the continuity terms produces:

$$(\rho \vec{\nabla})_t = \vec{\mathcal{M}} + \left[(\vec{\nabla} - 2a\hat{b}) \right] \left\{ \frac{\hat{b} \cdot [(\rho \vec{\nabla})_t - \vec{\mathcal{M}}]}{[\hat{b} \cdot \vec{\nabla} - 2a]} \right\} \quad (\text{D.33})$$

Applying the boundary condition and rearranging equation (D.33) leaves:

$$(\rho \vec{\nabla})_t = \vec{\mathcal{M}} + \left[\frac{\vec{\nabla}}{2a} - \hat{b} \right] (\hat{b} \cdot \vec{\mathcal{M}}) \quad (\text{D.34})$$

Equation (D.34) can be broken into scalar components. The x direction

component of equation (D.34) is:

$$(\rho u)_t = \mathcal{M}_i + \left[\frac{u}{2a} - b_i \right] (\hat{b} \cdot \vec{\mathcal{M}}) \quad (D.35)$$

where \mathcal{M}_i is the x direction component of the vector momentum equation space derivatives. Equation (D.35) is solved for the x direction component of momentum, ρu , at the solid blade surface boundary point.

Once the x direction component of momentum is known at the new time level, the y direction component of momentum is calculated using the boundary condition. Specifically, equation (D.22) is solved based on the local coordinate system at the boundary point in question. Equation (D.22) is repeated here for convenience:

$$\rho v = \left(\frac{c_j}{c_i} \right) \rho u \quad (D.36)$$

The last equation solved at the blade surface boundary point is based on the pathline equation, equation (D.14). Solving equation (D.14) for the temporal energy derivative term yields:

$$(\rho e)_t = \mathcal{E} - \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] (\rho_t - \mathcal{E}) + \vec{\nabla} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (D.37)$$

Using equations (D.29) and (D.34) to eliminate the continuity and vector momentum terms respectively, incorporating the boundary condition, and rearranging leaves:

$$(\rho e)_t = \mathcal{E} + \left[\frac{V^2}{4a} + \frac{a}{2(\gamma-1)} \right] (\hat{b} \cdot \vec{\mathcal{M}}) \quad (D.38)$$

Equation (D.38) is solved for the total energy, ρe , at the blade surface boundary point.

Equations (D.29), (D.35), (D.36), and (D.38) are solved at each blade surface boundary point to determine the solution at the new time level. Those four equations are repeated here:

$$\rho_t = \mathcal{E} + \left[\frac{1}{2a} \right] \hat{b} \cdot \vec{\mathcal{M}} \quad (\text{D.39})$$

$$(\rho u)_t = \mathcal{M}_i + \left[\frac{u}{2a} - b_i \right] (\hat{b} \cdot \vec{\mathcal{M}}) \quad (\text{D.40})$$

$$\rho v = \left[\frac{c_j}{c_i} \right] \rho u \quad (\text{D.41})$$

$$(\rho e)_t = \mathcal{E} + \left[\frac{V^2}{4a} + \frac{a}{2(\gamma-1)} \right] (\hat{b} \cdot \vec{\mathcal{M}}) \quad (\text{D.42})$$

The MacCormack method backward corrector calculations require predicted property values one row of grid points inside the solid boundary. This is achieved by extrapolating predicted values from the flowfield. Both linear and quadratic extrapolation have been used at the blade surface during the current research effort. With either approach some explicit smoothing is required to stabilize the solution. Using linear extrapolation produces good results. When quadratic extrapolation is used, the smoothing requirement is increased to such an extent that some significant flow features are smeared out. Therefore, linear extrapolation is used in the present investigation.

D.5 TRAILING EDGE POINT UNIT PROCESS

In the inviscid cascade flowfield, the Kutta condition is enforced at the trailing edge of the blade. Specifically, the flows leaving the two sides of the blade surface at the trailing edge are forced to flow parallel to each other in a direction such that the static pressure is equal on the two sides of the trailing edge point.

In the present investigation, the requirements listed above are enforced by installing a *small imaginary solid wall segment which is hinged at the trailing edge point*. The flow on each side of the hinged wall segment is solved independently, in a manner similar to the technique described in Section D.4 for the blade surface point. By forcing the flow on both sides of the wall segment to follow the wall, the flows on the two sides of the trailing edge are forced to be parallel to each other.

After solving for the flow properties on each side of the hinged wall segment independently, the pressure difference between the two sides is checked. If the difference is not equal to zero, to within a specified tolerance, a zero finding secant method is used to correct the wall orientation (flow angle) toward the angle where the pressures will be equal. This procedure is iterated until the pressure difference between the pressure and suction sides of the blade, at the trailing edge, is approximately zero.

As mentioned above, the flow on each side of the imaginary hinged wall segment is solved in a manner similar to the grid points which lie on the remainder of the blade surface. However, because the imaginary wall segment

is not stationary, but instead is allowed to pivot, the local coordinate systems, set up on each side of the wall at the trailing edge points, also pivot. This is illustrated in Figure D-2.

Throughout the trailing edge point unit process derivation, unless otherwise specified, the local coordinate system unit vectors, \hat{b} and \hat{c} , are defined with respect to their orientation at the forward time level, that is, the time level of the predicted and corrected properties. Therefore, unless otherwise specified, during the iterative solution process, \hat{b} is aligned perpendicular to, and \hat{c} is aligned parallel to the hinged wall segment at the angle the wall is assumed to be at, at the forward time level, during that iteration.

The boundary condition applicable at the trailing edge point imaginary hinged wall segment is that the velocity normal to the wall segment, at the hinge, at its forward time level, is zero.

$$\hat{b} \cdot \vec{V} = 0 \quad (D.43)$$

Multiplication by the density yields an equivalent boundary condition.

$$\hat{b} \cdot (\rho \vec{V}) = 0 \quad (D.44)$$

The momentum terms and the local coordinate system change with time. Therefore, taking the derivative of equation (D.44) with respect to time yields:

$$\hat{b}_t \cdot (\rho \vec{V}) + \hat{b} \cdot (\rho \vec{V})_t = 0 \quad (D.45)$$

Expanding equation (D.44) to scalar form and rearranging provides an expression for the y-momentum, ρv , in terms of the x-momentum, ρu .

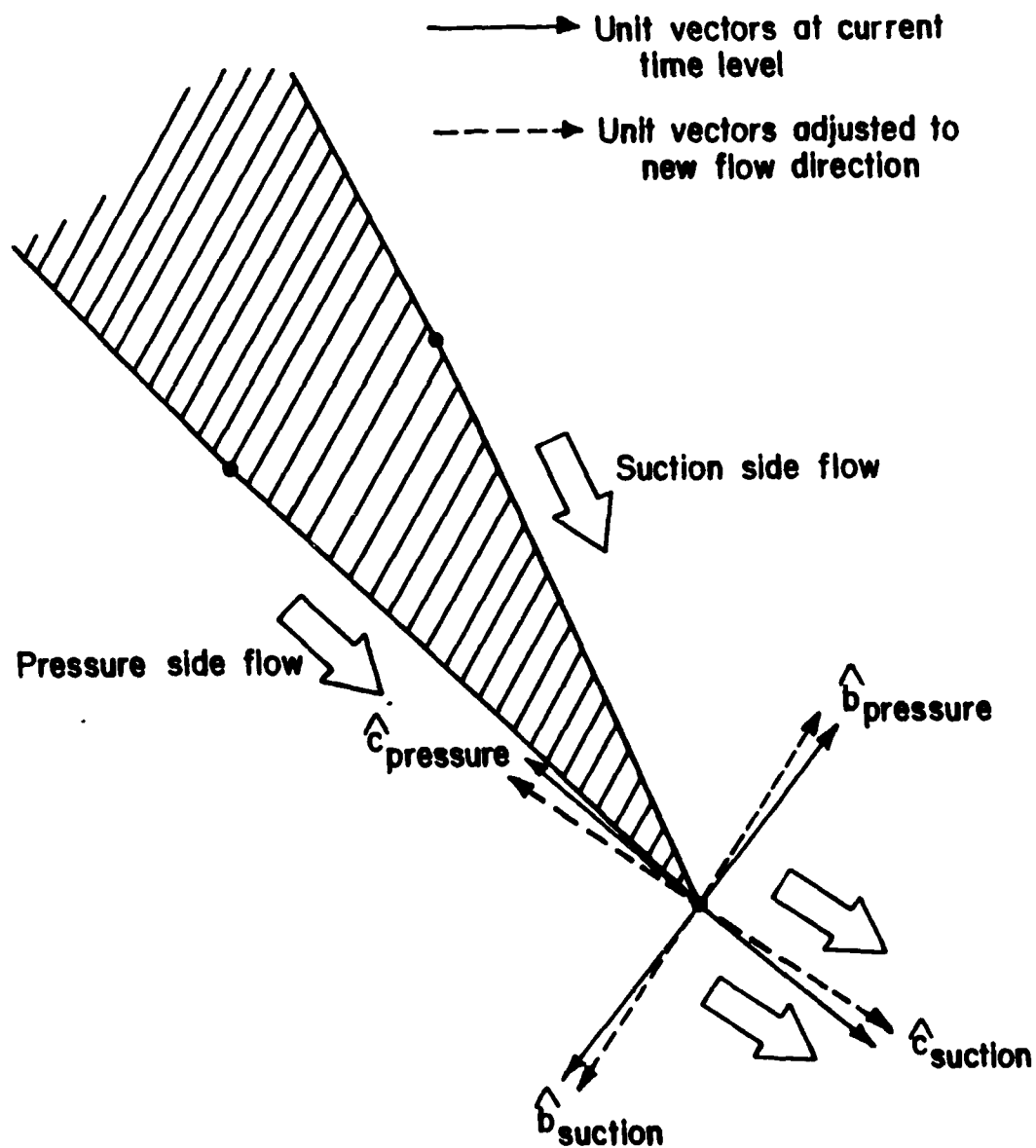


Figure D-2. Trailing edge point unit vectors and flow directions.

$$\rho v = - \left(\frac{b_i}{b_j} \right) \rho u = \left(\frac{c_j}{c_i} \right) \rho u \quad (D.46)$$

Equation (D.46) is true at any time as long as the unit vectors and the velocity are all defined for the same time level.

Later in this development, a temporal version of equation (D.46) is required. Taking the derivative of equation (D.46) with respect to time yields:

$$(\rho v)_t = \left(\frac{c_j}{c_i} \right)_t (\rho u) + \left(\frac{c_j}{c_i} \right) (\rho u)_t \quad (D.47)$$

To determine the applicable equations at the trailing edge point, the governing equations, equations (D.1) to (D.3), are replaced with compatibility equations, equations (D.14) and (D.15). The waveline equation, equation (D.15), is applied in three wave surfaces corresponding to the wave surface unit normal vectors \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (D.16) to (D.18). Since $\hat{n}_1 = \hat{b}$, the waveline equation corresponding to vector \hat{n}_1 extends beyond the hinged wall segment and therefore, does not affect the flow on the side of the wall under consideration. As at the blade surface boundary condition, the \hat{n}_1 waveline is intentionally placed outside the range of influence of the flow, and is replaced by the boundary condition, equation (D.43). Thus, the applicable equations at the trailing edge point are the boundary condition, equation (D.43), the pathline equation, equation (D.14), and the waveline equation, equation (D.15), applied in the remaining two wave surfaces corresponding to wave surface unit normal vectors \hat{n}_2 and \hat{n}_3 . Thus,

$$\hat{\mathbf{b}} \cdot \vec{\mathbf{V}} = 0 \quad (\text{D.48})$$

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{\mathbf{V}} \cdot (\rho \vec{\mathbf{V}})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{\mathbf{V}} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (\text{D.49})$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{\mathbf{V}}}{\gamma-1} \right] \rho_t - \left[\vec{\mathbf{V}} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot (\rho \vec{\mathbf{V}})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{\mathbf{V}}}{\gamma-1} \right] \mathcal{E} - \left[\vec{\mathbf{V}} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (\text{D.50})$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{\mathbf{V}}}{\gamma-1} \right] \rho_t - \left[\vec{\mathbf{V}} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot (\rho \vec{\mathbf{V}})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_3 \cdot \vec{\mathbf{V}}}{\gamma-1} \right] \mathcal{E} - \left[\vec{\mathbf{V}} + \frac{a \hat{n}_3}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (\text{D.51})$$

Equations (D.48) to (D.51) comprise an appropriate set of equations for the flows on the two sides of the hinged wall segment (trailing edge point). However, these equations are unnecessarily complicated. Equations (D.49) to (D.51) each contain more than one time derivative. These four equations can be rearranged into a form which is simpler and more computationally efficient when solved by a finite difference method. The rearrangement and simplification of this set of equations is demonstrated next.

Adding the two waveline equations, equations (D.50) and (D.51), and rearranging yields:

$$\left[V^2 + \frac{a(\hat{n}_2 + \hat{n}_3) \cdot \vec{V}}{\gamma - 1} \right] (\rho_t - \mathcal{E}) - \left[2\vec{V} + \frac{a(\hat{n}_2 + \hat{n}_3)}{\gamma - 1} \right] \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] + 2[(\rho e)_t - \mathcal{E}] = 0 \quad (D.52)$$

Subtracting twice the pathline equation, equation (D.49), from equation (D.52), using the result that $\hat{n}_2 + \hat{n}_3 = -\hat{b}$, and rearranging yields:

$$\left[\hat{b} \cdot \vec{V} - 2a \right] (\rho_t - \mathcal{E}) = \hat{b} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (D.53)$$

Solving for the temporal derivative of density leaves:

$$\rho_t = \mathcal{E} + \frac{\hat{b} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}]}{\left[\hat{b} \cdot \vec{V} - 2a \right]} \quad (D.54)$$

The boundary condition can not be used to cancel terms in equation (D.54) as in equation (D.29) because, if the imaginary wall segment has pivoted, the velocity vector at the previous time step will not be parallel to the wall at its new orientation. In other words, $\hat{b}_{\text{new}} \cdot \vec{V}_{\text{old}} \neq 0$.

Equation (D.54) is solved for the density on each side of the hinged wall segment (trailing edge point) to advance the solution in time. However, prior to solving equation (D.54), the temporal derivatives of momentum must be calculated.

The first step in deriving the equations which will be used to calculate the momentum components is to find another expression for the term $(\rho_t - \mathcal{E})$. Subtracting the \hat{n}_3 waveline equation, equation (D.51), from the \hat{n}_2 waveline equation, equation (D.50), recognizing that $\hat{n}_2 - \hat{n}_3 = \sqrt{3} \hat{c}$, and simplifying the

result produces:

$$(\hat{c} \cdot \vec{V})(\rho_t - \mathcal{E}) = \hat{c} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (\text{D.55})$$

Adding \hat{b} times equation (D.53) to \hat{c} times equation (D.55) yields:

$$(\rho \vec{V})_t - \vec{\mathcal{M}} = \left[\hat{b}(\hat{b} \cdot \vec{V} - 2a) + \hat{c}(\hat{c} \cdot \vec{V}) \right] (\rho_t - \mathcal{E}) \quad (\text{D.56})$$

Simplifying equation (D.56) leaves:

$$(\rho \vec{V})_t = \vec{\mathcal{M}} + \left[(\vec{V} - 2a\hat{b}) \right] (\rho_t - \mathcal{E}) \quad (\text{D.57})$$

Expanding equation (D.57) into its scalar components yields:

$$(\rho u)_t = \mathcal{M}_i + (u - 2ab_i)(\rho_t - \mathcal{E}) \quad (\text{D.58})$$

and

$$(\rho v)_t = \mathcal{M}_j + (v - 2ab_j)(\rho_t - \mathcal{E}) \quad (\text{D.59})$$

In order to eliminate the continuity terms from these equations, define the

scalar $g = \frac{u - 2ab_i}{v - 2ab_j}$. Then subtract g times equation (D.59) from equation (D.58).

$$(\rho u)_t - g(\rho v)_t = \mathcal{M}_i - g \mathcal{M}_j \quad (\text{D.60})$$

Using equation (D.47) to eliminate $(\rho v)_t$ from equation (D.60) produces:

$$(\rho u)_t - g \left[\left(\frac{c_j}{c_i} \right)_t (\rho u) + \left(\frac{c_j}{c_i} \right) (\rho u)_t \right] = \mathcal{M}_i - g \mathcal{M}_j \quad (\text{D.61})$$

Solving equation (D.61) for the $(\rho u)_t$ term yields:

$$(\rho u)_t = \frac{\mathcal{M}_i + g \left[-\mathcal{M}_j + \left(\frac{c_j}{c_i} \right)_t (\rho u) \right]}{1 - g \left(\frac{c_j}{c_i} \right)} \quad (\text{D.62})$$

Equation (D.62) is solved for the x direction component of momentum on each side of the imaginary hinged wall segment at the blade trailing edge point to advance the solution in time. At the predictor step of the MacCormack method, the current time level values of the fluid properties and of the \hat{c} vector are used. At the corrector step of the MacCormack method, the predicted property values and the corresponding \hat{c} vector are used. Obviously, calculation of the temporal derivative of the ratio of the components of \hat{c} requires \hat{c} components at the current time level and assumed \hat{c} components at the forward time level.

Once the x direction component of momentum is known at the new time level, the y direction component of momentum is calculated using the boundary condition. Specifically, equation (D.46) is solved based on the local coordinate system at the forward time level. Equation (D.46) is repeated here for convenience.

$$\rho v = \left(\frac{c_j}{c_i} \right) \rho u \quad (\text{D.63})$$

The last equation solved at the trailing edge boundary point in the present investigation is based on the pathline equation, equation (D.14). Solving equation (D.14) for the temporal energy term yields:

$$(\rho e)_t = \mathcal{E} - \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] (\rho_t - \mathcal{E}) + \vec{V} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (D.64)$$

Using equation (D.53) to eliminate the continuity term and rearranging leaves:

$$(\rho e)_t = \mathcal{E} + \left[\vec{V} + \left\{ \frac{\frac{V^2}{2} - \frac{a^2}{\gamma-1}}{2a - \hat{b} \cdot \vec{V}} \right\} \hat{b} \right] \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (D.65)$$

Equation (D.65) is solved for the energy, ρe , at the blade surface boundary point to advance the solution in time.

Solving equations (D.54) and (D.65) requires temporal derivatives of momentum. Therefore, equations (D.62) and (D.63) are solved, then the temporal derivatives of momentum are calculated numerically and substituted into equations (D.54) and (D.65).

In summary, equations (D.54), (D.62), (D.63), and (D.65) are solved on each side of the imaginary hinged wall segment located at the blade trailing edge to advance the solution in time. The solution process is iterated, changing the flow angle at each iteration, until the pressures on the two sides of the trailing edge are equal to within a specified tolerance. For the results presented in Section VII, a tolerance of 10^{-6} is used to evaluate convergence of the difference in the two values of pressure normalized by the inflow total pressure. Equations (D.54), (D.62), (D.63), and (D.65) are repeated here for convenience:

$$\rho_t = \mathcal{E} + \frac{\hat{b} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}]}{[\hat{b} \cdot \vec{V} - 2a]} \quad (D.66)$$

$$(\rho u)_t = \frac{\mathcal{M}_i + g \left[-\mathcal{M}_j + \left(\frac{c_j}{c_i} \right)_t (\rho u) \right]}{1 - g \left(\frac{c_j}{c_i} \right)} \quad (\text{D.67})$$

$$\rho v = \left(\frac{c_j}{c_i} \right) \rho u \quad (\text{D.68})$$

$$(\rho e)_t = \mathcal{E} + \left[\vec{V} + \left\{ \frac{\frac{V^2}{2} - \frac{a^2}{2}}{2a - \hat{b} \cdot \vec{V}} \right\} \hat{b} \right] \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (\text{D.69})$$

where $g = (u - 2ab_i)/(v - 2ab_j)$. In the denominator of equation (D.67), the \hat{c} vector is parallel to the hinged wall segment at the current time level for the predictor calculations, and is parallel to the hinged wall segment at its assumed position at the forward time level for the corrector calculations. For both steps, the temporal derivative of the ratio of the \hat{c} components is equal to the ratio at the forward time level, minus the ratio at the current time level, all divided by the time step.

The same set of equations must be solved at the predictor and the corrector steps of the MacCormack method. Therefore, the flow angle, and the corresponding forward time level definitions of \hat{b} and \hat{c} , must be the same for both steps. This constrains the iteration sequence to include both the predictor and corrector steps. Therefore, all points on the C-grid except the trailing edge point are predicted, next the trailing edge point is iteratively predicted and corrected, then all remaining points are corrected.

D.6 SUBSONIC INFLOW BOUNDARY POINT UNIT PROCESS

When the component of the fluid velocity perpendicular to the inflow boundary is less than the local speed of sound, the fluid properties at that boundary are dependent upon both upstream and downstream phenomena. If the fluid velocity were zero, the fluid properties would be influenced from all directions equally. When the velocity is not negligible, a greater influence will be felt from the upstream direction. From a characteristic perspective, this means that most of the base of the Mach cone (domain of dependence) lies outside the computational domain at the subsonic inflow boundary.

At an inflow boundary, one to four boundary conditions can be set. It would only be appropriate to set four boundary conditions if the flow component crossing the inflow boundary were supersonic, and therefore, the flow properties were totally dependent upon the upstream conditions. Setting three boundary conditions implies that a majority, but not all of the flow influences are coming from the upstream direction. For the present investigation, three boundary conditions are set at the inflow boundary.

The boundary conditions chosen for the subsonic inflow boundary are those which reflect the properties of the flow which are most likely to be known upstream of a turbine blade row. Specifically, the stagnation pressure, P_0 , the stagnation temperature, T_0 , and the flow angle, θ , are specified as boundary conditions at the inflow boundary.

To determine the applicable equations at the subsonic inflow boundary, the governing equations, equations (D.1) to (D.3), are replaced by compatibility

equations, equations (D.14) and (D.15). The waveline equation, equation (D.15), is applied in three wave surfaces corresponding to the wave surface unit normal vectors \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (D.16) to (D.18). At the inflow boundary the unit vector \hat{b} is oriented perpendicular to the inflow boundary and pointing in. This situation is illustrated in Figure D-1. Thus, the wavelines corresponding to the \hat{n}_2 and \hat{n}_3 vectors and the pathline, which are all outside the computational domain, are replaced by the three boundary conditions. Only the waveline corresponding to the \hat{n}_1 unit vector is placed inside the computational domain. Thus, the applicable equations at the subsonic inflow boundary are the three boundary conditions, P_0 , T_0 , and θ , and the waveline equation applied along the \hat{n}_1 unit vector.

$$P_0 = P_{0_{\text{inflow}}} \quad (\text{D.70})$$

$$T_0 = T_{0_{\text{inflow}}} \quad (\text{D.71})$$

$$\theta = \theta_{\text{inflow}} \quad (\text{D.72})$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a\hat{n}_1 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a\hat{n}_1}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ & = \left[\frac{V^2}{2} + \frac{a\hat{n}_1 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a\hat{n}_1}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (\text{D.73})$$

Equations (D.70) to (D.73) comprise an appropriate set of equations for the inflow boundary. In previous unit process derivations, multiple compatibility equations made it possible to use linear algebra to simplify the applicable

equations prior to implementation. Since only one compatibility equation is used at the subsonic inflow boundary, it must be computed without simplification. This means that all the temporal derivatives in equation (D.73) must be evaluated simultaneously. This is achieved through iteration.

The Mach number, M , is the iteration variable in the subsonic inflow boundary point unit process. Knowing an estimated forward time Mach number and the boundary conditions given by equations (D.70) to (D.72), several additional properties can be computed. In this unit process, the density, ρ , the x-direction and y-direction components of momentum, ρu and ρv , and the energy, ρe , are needed at the forward time for the iterative process. Therefore, in terms of known properties of the flow and the Mach number:

$$\rho = \frac{P_0}{RT_0} \left(\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right)^{\frac{1}{\gamma-1}} \quad (D.74)$$

$$\rho u = P_0 M \left[\frac{\gamma}{RT_0 \left(1 + \tan^2(\theta) \right) \left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma+1}{\gamma-1}}} \right]^{\frac{1}{2}} \quad (D.75)$$

$$\rho v = \rho u \tan(\theta) \quad (D.76)$$

$$\rho e = P_0 \left[\frac{\gamma}{\gamma-1} - \frac{1}{1 + \frac{\gamma-1}{2} M^2} \right] \left[\frac{1}{1 + \frac{\gamma-1}{2} M^2} \right]^{\frac{1}{\gamma-1}} \quad (D.77)$$

The iterative process used to find property values at subsonic inflow boundary points is described next.

For both predictor and corrector steps, the iterative procedure is initiated by guessing the forward time Mach number at the inflow grid point being considered. For the first time step, the Mach number guess is calculated from the property values at the initial-value surface. Subsequently, the initial Mach number guess is calculated from the property values at the current time level. The guessed Mach number is used in equations (D.74) to (D.77) to calculate guessed forward time values of ρ , ρu , ρv , and ρe .

Equation (D.73) is rearranged to solve for the temporal derivative of energy:

$$(\rho e)_t = \mathcal{E} - \left[\frac{V^2}{2} + \frac{\mathbf{a}\hat{\mathbf{b}} \cdot \vec{V}}{\gamma - 1} \right] (\rho_t - \mathcal{E}) + \left[\vec{V} + \frac{\mathbf{a}\hat{\mathbf{b}}}{\gamma - 1} \right] [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (\text{D.78})$$

Temporal derivatives of density and momentum are calculated numerically using the current and guessed forward time values of these properties. Using these temporal derivatives and the transformed space derivatives, equation (D.78) is solved for a calculated value of ρe . If the calculated and guessed values of ρe are equal, to within a specified tolerance, the Mach number guess was correct and the four guessed forward time property values are the predicted (or corrected) property values at the new time step. If the calculated and guessed values of ρe are not approximately equal, a zero finding secant method is used to home in on the correct Mach number. For the results presented in Section VII, a tolerance of 10^{-6} is used to evaluate

convergence of the energy difference normalized by the value of energy at the current time level.

For the predictor step, forward differenced space derivatives are required. Therefore, quadratic extrapolation is used to approximate property values outside the computational domain.

D.7 EXIT BOUNDARY POINT UNIT PROCESSES

Because the component of the flow velocity which crosses the exit boundary can be either subsonic or supersonic, each exit boundary point must be checked, at each time step, to determine which unit process is appropriate. At the exit boundary, the unit vector \hat{b} is placed perpendicular to the boundary, pointing outward, as illustrated in Figure D-1. Therefore, the quantity $\hat{b} \cdot \vec{V}$ is calculated and compared to the local speed of sound, a , to determine whether the boundary should be handled as a supersonic or a subsonic exit boundary.

When the component of the fluid velocity perpendicular to the exit boundary is less than the local speed of sound, the fluid properties at that boundary are dependent upon both upstream and downstream phenomena. Therefore, at least one boundary condition must be applied to the solution procedure. This situation is discussed in Section D.7.1 of this appendix.

When the component of the fluid velocity perpendicular to the exit boundary is greater than or equal to the local speed of sound, the fluid mechanics at the exit are only dependent upon upstream conditions. From a characteristic perspective, this means that the base of the Mach cone (domain

of dependence) lies entirely within the computational domain. In this case the interior point unit process is appropriate at an exit boundary point. This situation is discussed in Section D.7.2 of this appendix.

On a C-grid, the exit boundary is a single straight line with a length equal to the blade spacing. However, the two ends of the computational grid ($\xi = 1 = \text{constant}$ and $\xi = \xi_{\max} = \text{constant}$ grid lines) make up the exit boundary. This distinction does not affect the derivation of applicable equations at the exit boundary. However, these two sections must be implemented in a slightly different manner. Specifically, the numerical approximation of space derivatives in the ξ direction requires property value extrapolations beyond the exit at the predictor step for the $\xi = \xi_{\max}$ boundary, and at the corrector step for the $\xi = 1$ boundary. In both cases, quadratic extrapolation is used.

D.7.1 SUBSONIC EXIT BOUNDARY POINT UNIT PROCESS. The boundary condition applied at a subsonic exit boundary point is that the exit static pressure is known:

$$P = P_{\text{exit}} \quad (\text{D.79})$$

Because pressure is not one of the four solution variables being computed at each grid point, the solution procedure at the subsonic exit is iterative.

To determine the applicable equations at the subsonic exit boundary point, the governing equations, equations (D.1) to (D.3), are replaced by compatibility equations, equations (D.14) and (D.15). The waveline equation, equation

(D.15), is applied in three wave surfaces corresponding to the wave surface unit normal vectors \hat{n}_1 , \hat{n}_2 , and \hat{n}_3 , given by equations (D.16) to (D.18). Since the boundary condition, equation (D.79), must be incorporated, the waveline equation corresponding to vector \hat{n}_1 (which was intentionally placed outside the flowfield) is replaced by the boundary condition. This situation is illustrated in Figure D-1. The applicable equations at a subsonic exit boundary point are the boundary condition, equation (D.79), the pathline equation, equation (D.14), and the waveline equation, equation (D.15), applied in the remaining two wave surfaces corresponding to wave surface unit normal vectors \hat{n}_2 and \hat{n}_3 :

$$P = P_{\text{exit}} \quad (\text{D.80})$$

$$\left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \rho_t - \vec{V} \cdot (\rho \vec{V})_t + (\rho e)_t = \left[\frac{V^2}{2} - \frac{a^2}{\gamma-1} \right] \mathcal{E} - \vec{V} \cdot \vec{\mathcal{M}} + \mathcal{E} \quad (\text{D.81})$$

$$\begin{aligned} & \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\ &= \left[\frac{V^2}{2} + \frac{a \hat{n}_2 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a \hat{n}_2}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E} \end{aligned} \quad (\text{D.82})$$

$$\begin{aligned}
& \left[\frac{V^2}{2} + \frac{a\hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \rho_t - \left[\vec{V} + \frac{a\hat{n}_3}{\gamma-1} \right] \cdot (\rho \vec{V})_t + (\rho e)_t \\
& = \left[\frac{V^2}{2} + \frac{a\hat{n}_3 \cdot \vec{V}}{\gamma-1} \right] \mathcal{E} - \left[\vec{V} + \frac{a\hat{n}_3}{\gamma-1} \right] \cdot \vec{\mathcal{M}} + \mathcal{E}
\end{aligned} \tag{D.83}$$

Equations (D.80) to (D.83) comprise an appropriate set of equations for the subsonic exit boundary point. However, these equations are unnecessarily complicated. Equations (D.81) to (D.83) each contain more than one time derivative. Equations (D.81) to (D.83) can be rearranged into a form which is simpler and more computationally efficient to solve by a finite difference method. The rearrangement and simplification of this set of equations is demonstrated next.

Adding the two waveline equations, equations (D.82) and (D.83), and rearranging yields:

$$\begin{aligned}
& \left[V^2 + \frac{a(\hat{n}_2 + \hat{n}_3) \cdot \vec{V}}{\gamma-1} \right] (\rho_t - \mathcal{E}) \\
& - \left[2\vec{V} + \frac{a(\hat{n}_2 + \hat{n}_3)}{\gamma-1} \right] \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] + 2[(\rho e)_t - \mathcal{E}] = 0
\end{aligned} \tag{D.84}$$

Subtracting twice the pathline equation, equation (D.81), from equation (D.84), using the result that $\hat{n}_2 + \hat{n}_3 = -\hat{b}$, and simplifying yields:

$$\left[\hat{b} \cdot \vec{V} - 2a \right] (\rho_t - \mathcal{E}) = \hat{b} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \tag{D.85}$$

Subtracting the \hat{n}_3 waveline equation, equation (D.83), from the \hat{n}_2 waveline equation, equation (D.82), recognizing that $\hat{n}_2 - \hat{n}_3 = \sqrt{3} \hat{c}$, and simplifying the result produces:

$$(\hat{c} \cdot \vec{V})(\rho_t - \mathcal{E}) = \hat{c} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] \quad (\text{D.86})$$

Adding \hat{b} times equation (D.85) to \hat{c} times equation (D.86) yields:

$$(\rho \vec{V})_t - \vec{\mathcal{M}} = \left[\hat{b}(\hat{b} \cdot \vec{V} - 2a) + \hat{c}(\hat{c} \cdot \vec{V}) \right] (\rho_t - \mathcal{E}) \quad (\text{D.87})$$

Simplifying equation (D.87) leaves:

$$(\rho \vec{V})_t - \vec{\mathcal{M}} = \left[(\vec{V} - 2a\hat{b}) \right] (\rho_t - \mathcal{E}) \quad (\text{D.88})$$

The vector dot product of the velocity vector and equation (D.88) is:

$$\vec{V} \cdot [(\rho \vec{V})_t - \vec{\mathcal{M}}] = \left[V^2 - 2a\hat{b} \cdot \vec{V} \right] (\rho_t - \mathcal{E}) \quad (\text{D.89})$$

Adding equation (D.89) to the pathline equation and rearranging yields:

$$\rho_t = \mathcal{E} + \frac{(\rho e)_t - \mathcal{E}}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2a\hat{b} \cdot \vec{V}} \quad (\text{D.90})$$

Equation (D.90) is solved for density at subsonic exit boundary points to advance the solution in time. The value of $(\rho e)_t$ is determined during the iterative solution process. This process will be discussed later.

Combining equations (D.90) and (D.88) to eliminate the continuity terms yields:

$$(\rho \vec{V})_t = \vec{\mathcal{M}} + \frac{[(\rho e)_t - \mathcal{E}] [\vec{V} - 2a\hat{b}]}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2a\hat{b} \cdot \vec{V}} \quad (\text{D.91})$$

Expanding equation (D.91) into scalar components produces:

$$(\rho u)_t = \mathcal{M}_i + \frac{[(\rho e)_t - \mathcal{E}] [u - 2a b_i]}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2a \hat{b} \cdot \vec{V}} \quad (D.92)$$

$$(\rho v)_t = \mathcal{M}_j + \frac{[(\rho e)_t - \mathcal{E}] [v - 2a b_j]}{\frac{a^2}{\gamma-1} + \frac{V^2}{2} - 2a \hat{b} \cdot \vec{V}} \quad (D.93)$$

Equations (D.92) and (D.93) are solved for the components of momentum at subsonic exit boundary points to advance the solution in time. The value of $(\rho e)_t$ is determined during the iterative solution process.

Up to this point the boundary condition, equation (D.80), has not been incorporated. In addition, no equation has been derived to determine the value of the energy at a subsonic exit boundary point. To tie these two factors together, consider the energy. Since body forces are negligible, $\rho e = \hat{u} + \frac{1}{2}\rho V^2$. Since the fluid is assumed to be a perfect gas, this relationship becomes: $\rho e = \frac{P}{\gamma-1} + \frac{1}{2}\rho V^2$. Solving this equation for the static pressure leaves:

$$P = (\gamma-1)[\rho e - \frac{1}{2}\rho V^2] \quad (D.94)$$

This static pressure must be the specified pressure at the subsonic exit.

The following iterative process is used to calculate the flow properties at the subsonic exit points. The process is initiated by guessing a value for the energy, ρe , for the new time level. For the first time step, the energy value is obtained from the initial-value surface. Subsequently, the initial guess is set at the energy value of the current time level.

The current time level value of energy and the value of energy guessed for the forward time level are used to numerically calculate a value for the temporal derivative of energy. This temporal derivative is used in equations (D.90), (D.92), and (D.93) to calculate values for the density and the components of momentum. These values of density and momentum and the guessed forward time value of energy are used in equation (D.94) to calculate a value for pressure. If the calculated value of pressure is equal to the specified exit pressure, to within a specified tolerance, the energy value guess is correct and the calculated values of ρ , ρu , ρv , and the guessed value of ρe are the property values at the forward time. If the calculated value of pressure does not agree with the specified exit pressure, a zero finding secant method is used to home in on the correct value of ρe . For the results presented in Section VII, a tolerance of 10^{-6} is used to evaluate convergence of the pressure difference.

D.7.2 SUPERSONIC EXIT BOUNDARY POINT UNIT PROCESS. At an exit boundary point, when the component of the fluid velocity perpendicular to the exit boundary is greater than or equal to the local speed of sound, the fluid mechanics at the exit are influenced only by upstream conditions. In that case the interior point unit process is appropriate. Therefore, the MacCormack method is applied directly to the transformed governing equations in the strong conservation-law form, equation (2.33). The resulting finite difference equations are used to advance the solution in time.

D.8 GRID CUT POINTS

All points on the grid cut lie in one of the regions discussed above. Specifically, the trailing edge of the blade, one exit point, and several interior points make up the grid cut on a C-type grid. Therefore, no new equations need to be derived for the points on the grid cut. However, two factors complicate the solution at points on the grid cut.

1. Each physical grid point on the grid cut is represented by two noncoincident grid points on the computational grid.
2. In the computational space, where the calculations are carried out and where flow property values are stored, there are only grid points in the positive η direction from the grid cut. Therefore, the MacCormack method corrector calculation can not be carried out without special indexing.

These issues are resolved for the trailing edge point in Section D.5 of this appendix.

The interior points are handled as described in Section D.3, with a few additional steps included to resolve the problems listed above. For every $\xi = \text{constant}$ grid line intersecting the lower side of the grid cut, the $(\xi_{\max} + 1 - \xi) = \text{constant}$ grid line intersects the same physical location on the upper side of the grid cut. All grid cut calculations are computed at the computational grid points which lie on the lower side of the grid cut. Therefore, the property values at the $(\xi, 1)$ and $(\xi, 2)$ grid points are used to predict the η derivatives at the grid cut points. Then, the predicted property

values at the $(\xi, 1)$ and $(\xi_{\max}+1-\xi, 2)$ grid points are used to correct the η derivatives at the grid cut points.

After the property values have been predicted at $(\xi, 1)$, these predicted values are transferred to the $(\xi_{\max}+1-\xi, 1)$ predictor values storage locations for use in the $(\xi_{\max}+1-\xi, 2)$ corrector calculations. After the corrector calculations have been completed, the corrected property values are transferred to the $(\xi_{\max}+1-\xi, 1)$ corrector value storage locations.

The exit point which lies on the grid cut is handled as described in Section D.7, but with the same additional indexing steps just described for the grid cut interior points.

D.9 PERIODIC BOUNDARY POINTS

All points on the periodic boundary lie in one of the regions discussed above. Specifically, one exit point and several interior points make up the periodic boundary on a C-type grid. Therefore, no new equations need to be derived for the points on the periodic boundary. However, three factors complicate the flow solutions at points on the periodic boundary.

1. Since the flow through a cascade is assumed to be exactly periodic, the property values at each point on the periodic boundary on the lower side of the grid must be exactly the same values found at the corresponding grid point on the periodic boundary on the upper side of the grid.
2. In the computational space, where the calculations are carried out and where flow property values are stored, there are grid points only in the

negative η direction from the periodic boundary. Therefore, the MacCormack method predictor calculation can not be carried out without special indexing.

3. At the periodic boundary grid point adjacent to the inflow boundary, there are two grid points upstream of this point along $\eta = \text{constant}$ grid lines. Therefore, a decision must be made concerning how to calculate the ξ space derivatives at these leading periodic boundary grid points.

The periodic boundary interior points are handled as described in Section D.3 with a few additional steps included to resolve the problems listed above. For every $\xi = \text{constant}$ grid line intersecting the the periodic boundary on the lower side of the grid, the $(\xi_{\text{max}}+1-\xi) = \text{constant}$ grid line is the corresponding grid line on the upper side of the grid. All periodic boundary calculations are computed at the computational grid points which lie on the periodic boundary on the lower side of the grid. Therefore, the property values at the (ξ, η_{max}) and $(\xi_{\text{max}}+1-\xi, \eta_{\text{max}}-1)$ grid points are used to predict the η derivatives at the periodic boundary points. Then, the predicted property values at the (ξ, η_{max}) and $(\xi, \eta_{\text{max}}-1)$ grid points are used to correct the η derivatives at the periodic boundary points. After the corrector calculations have been completed, the corrected property values are transferred to the $(\xi_{\text{max}}+1-\xi, \eta_{\text{max}})$ corrector values storage locations.

The special periodic boundary points which lie adjacent to the inflow boundary are handled as discussed above with one additional consideration. Both predictor and corrector ξ space derivatives are calculated entirely on the

$\eta = \eta_{\max} = \text{constant}$ line on the lower side of the C-grid. In other words, property values at (ξ, η_{\max}) and $(\xi+1, \eta_{\max})$ are used to calculate the predictor ξ -direction derivatives at (ξ, η_{\max}) . Therefore, property values at $(\xi_{\max}-\xi, \eta_{\max})$ are not used to approximate property derivatives at this special point.

The exit point which lies on the periodic boundary is handled as described in Section D.7, but, with the same additional indexing steps just described for the normal interior points which lie on the periodic boundary.

APPENDIX E

GRID GENERATION

Two of the major problems facing an analyst when constructing the numerical solution of partial differential equations are: the numerical implementation of the boundary conditions along the boundaries of the physical space, and the selection of the finite difference mesh to represent the continuous physical space. The boundaries of the physical space do not generally lie along coordinate lines formed by an equally spaced orthogonal grid system. When first-order accuracy is acceptable, boundary conditions can be implemented along arbitrary lines in the physical space and/or computations can be carried out on variable meshes. However, when higher-order accuracy is necessary, the above factors present serious difficulties. This has led to the extensive use of coordinate transformations to map the boundaries of physical space onto coordinate lines of a transformed space and to map nonuniform, nonorthogonal grids in physical space into uniform, orthogonal grids in transformed space. Once computational and physical grids have been generated, the numerical solution is implemented using the appropriate transformed governing equations.

In the first part of this section some favorable characteristics of grid generation methods are described. Then, the elliptic partial differential equation grid generation method chosen for this effort is described in more detail. Third, the techniques used to distribute points around the cascade

blade grid boundaries are summarized. Lastly, two illustrative examples are presented to demonstrate additional capabilities of the grid generator.

E.1 BACKGROUND

The following list presents five features that are often required of grid generators. Additional features may be required depending upon the specific problem under consideration. For some simple problems, one or more of the features listed below may be unnecessary.

1. The grid in the computational (transformed) space must be an equally spaced orthogonal grid. Such a grid is required to enable the development of accurate finite difference approximations to the transformed governing equations which are to be solved numerically in the computational space.
2. The coordinate transformation must yield a unique, one-to-one, correspondence of all points. In other words, there is one and only one point in the computational space corresponding to each point in the physical space, and vice versa.
3. The coordinate transformation must be body-fitted. That is, the arbitrary boundaries in physical space must map directly onto the straight boundaries of the computational grid.
4. There must be no crossing of coordinate surfaces of the same family, or of any interior coordinate lines with the physical boundaries. In other words, both the minimum and maximum values of the transformed

coordinates must occur on the physical boundaries.

5. The transformation must be continuous, have a continuous inverse, and have a nonvanishing Jacobian. All physical coordinate lines and surfaces should form smooth curves in the grid interior.

Partial differential equation methods are flexible in their application and can fulfill all of the requirements listed above. When all the boundary points of a physical domain are to be specified, elliptic partial differential equation methods are appropriate. When some boundaries are unspecified (e.g., free-stream conditions "far from" an aircraft in an external flow calculation), parabolic or hyperbolic grid generators may be more appropriate.

In most cascade flow studies, the flow past any blade is assumed to be identical to the flow past every other blade. Therefore, periodic boundaries are chosen which isolate a control volume containing a representative turbine blade and its associated flowfield. Therefore, exact locations for all boundaries are known and are specified as Dirichlet boundary conditions. Elliptic partial differential equation grid generation techniques are used to determine the physical locations of grid points in the interior of the grid.

E.2 POISSON-TYPE CASCADE GRIDS

In this research, a Poisson-type elliptic partial differential equation grid generator is used. The Poisson equation is simply the Laplace equation with a nonhomogeneous source term added.

The Laplace equation, applied to the generic variable $f(x,y)$, is presented in equation (E.1):

$$\nabla^2 f = f_{xx} + f_{yy} = 0 \quad (\text{E.1})$$

Adding a source term to equation (E.1) yields the Poisson equation:

$$\nabla^2 f = f_{xx} + f_{yy} = F(x,y) \quad (\text{E.2})$$

Both the Laplace equation and the Poisson equation are elliptic partial differential equations. While a grid can be generated using the simpler Laplace equation, the freedom to specify the nonhomogeneous terms in the Poisson equation provides the added benefit of being able to control the relative point distribution in the interior of the grid. For example, coordinate surfaces can be clustered near a solid boundary. This capability is discussed in more detail in section E.2.1. When multidimensional grids of this type are generated, one Poisson equation is used for each space dimension. Thus, for a two-dimensional Poisson-type grid generator, the following equations must be solved numerically:

$$\nabla^2 \xi = \xi_{xx} + \xi_{yy} = P \quad (\text{E.3})$$

$$\nabla^2 \eta = \eta_{xx} + \eta_{yy} = Q \quad (\text{E.4})$$

where P and Q are the nonhomogeneous source terms.

As mentioned earlier, for a cascade flowfield periodic boundaries are chosen which isolate one period of the flow and allow the analyst to specify exact locations for each point on every boundary. Several general shapes of physical grids can be fit to a cascade blade. While a flow-through, or H-type grid, as

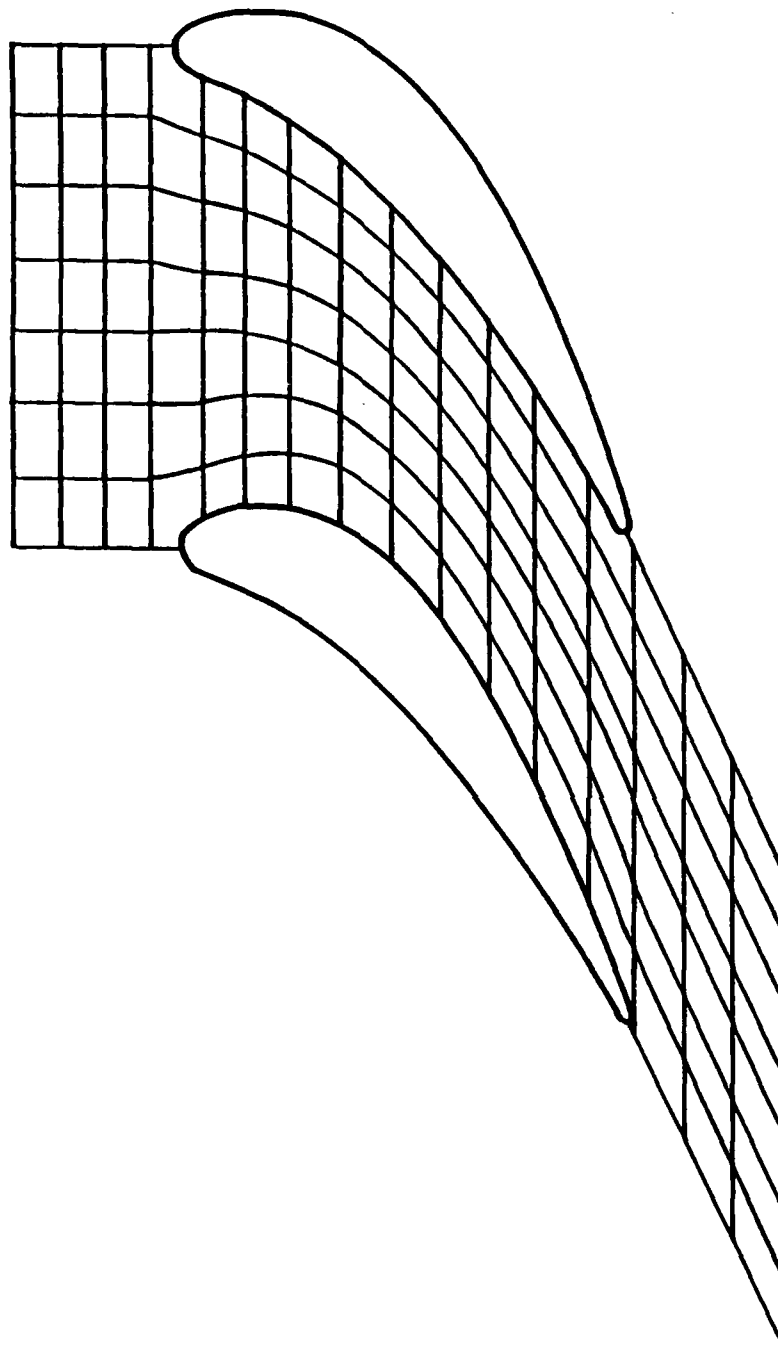


Figure E-1. H-type cascade blade grid.

shown in Figure E-1, is an obvious choice, these grids are seldom used since resolution is normally poor near the blade leading edge. O-type grids are grids in which the $\eta = \text{constant}$ grid lines form closed loops around the blade, as shown in Figure E-2. This type of grid is acceptable for the present study. However, no family of grid lines is generally aligned with the flow beyond the trailing edge. This can cause problems if the associated flow-solver is ever extended to include viscous calculations. In addition, O-type grids often contain awkward and oversized grid cells near the intersection of the periodic boundary and the exit boundary. These awkward cells can create relatively large truncation errors in the flow solution. Therefore, O-type grids are not used in this investigation.

C-type grids, like the one shown in Figure E-3, will be used in the present investigation. Obviously, the name is descriptive of the shape of each of the members of the family of grid lines that surround the blade. Figure E-4 shows the relationship between the boundaries of the C-grid and the boundaries of the corresponding computational grid.

Solving equations (E.3) and (E.4) numerically would be simple if the computations could be carried out on the physical grid. Unfortunately, due to unequal spacing in this grid, only first-order accuracy can be achieved using this approach. Therefore, the Poisson equations are transformed to the equally-spaced orthogonal computational domain and solved there. The transformed Poisson equations are:

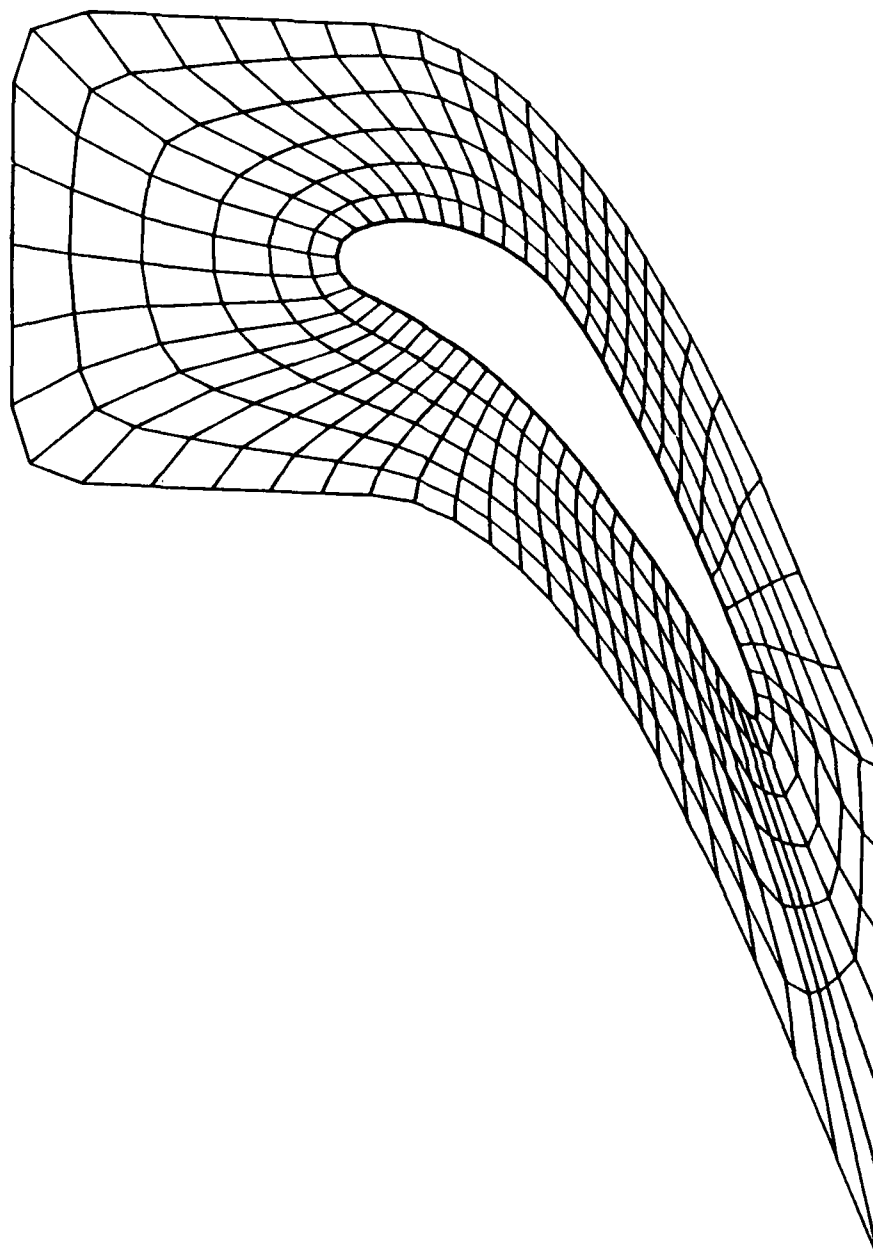


Figure E-2. O-type cascade blade grid.

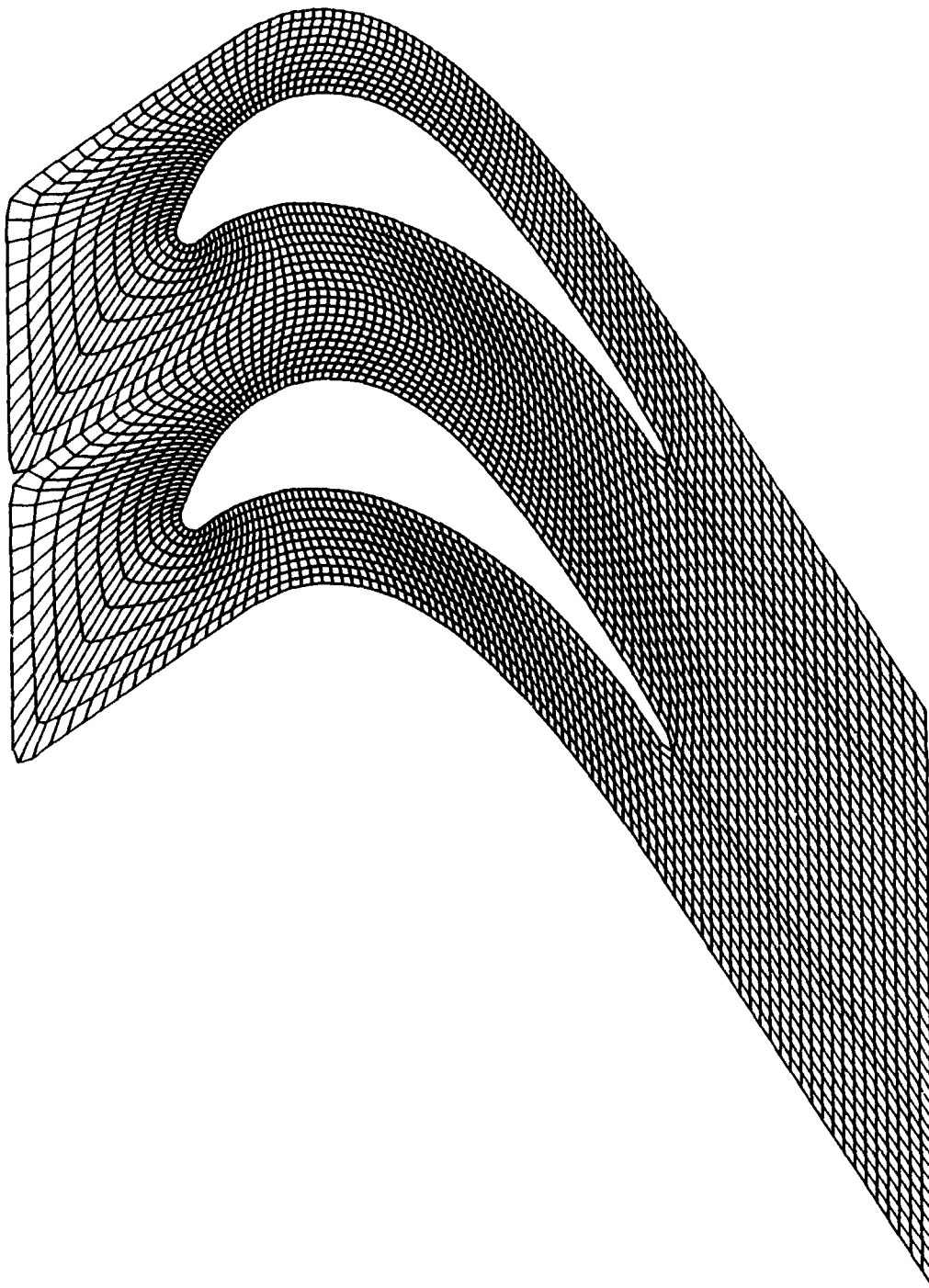


Figure E-3. C-type cascade blade grid.

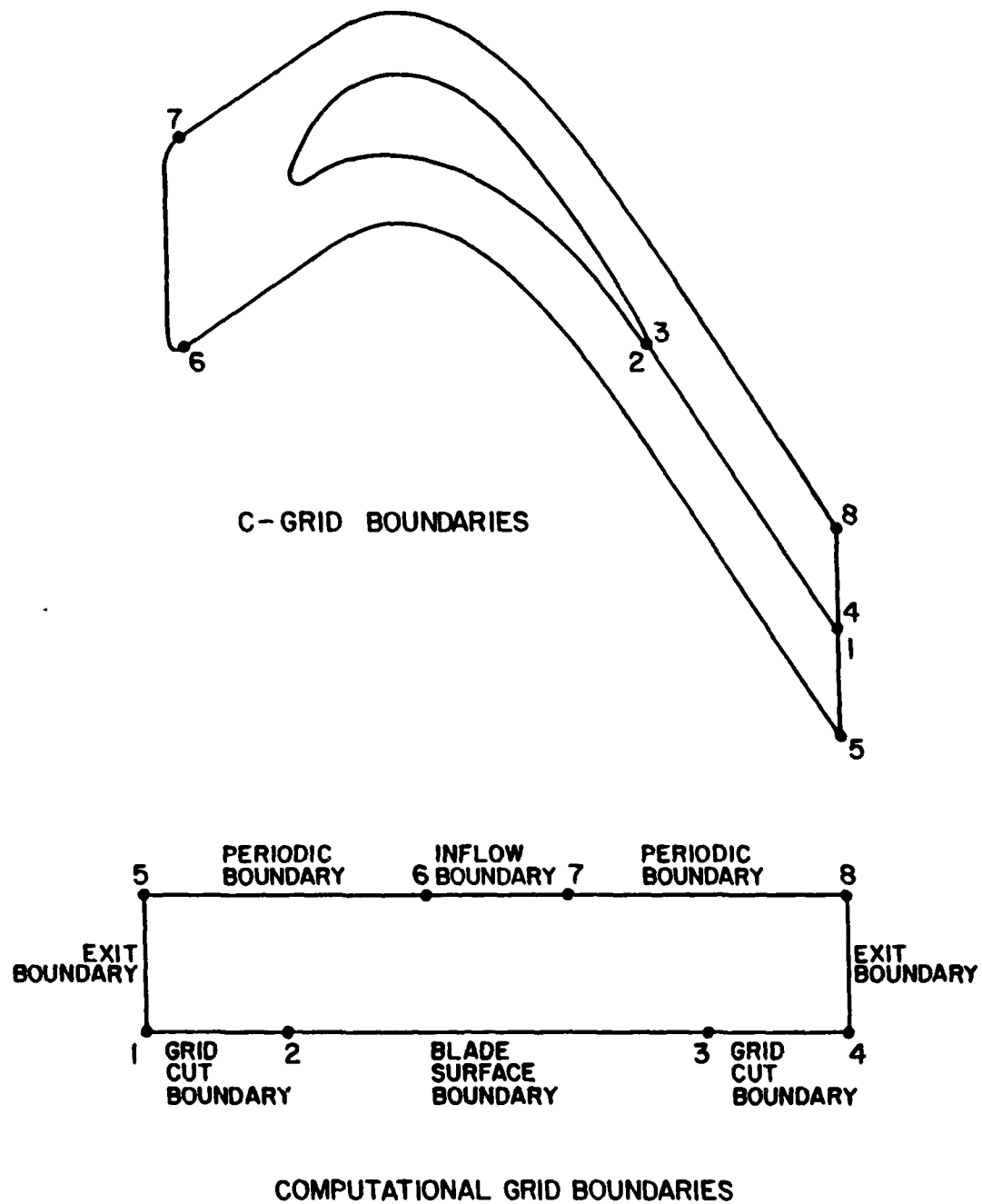


Figure E-4. C-type grid physical and computational boundaries.

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -I^2(Px_{\xi} + Qx_{\eta}) \quad (E.5)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -I^2(Py_{\xi} + Qy_{\eta}) \quad (E.6)$$

where

$$I = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}$$

$$\alpha = x_{\eta}^2 + y_{\eta}^2$$

$$\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2$$

For the C-type grid, grid points on the blade surface boundary, the grid cut extending downstream from the trailing edge, the inflow boundary, the periodic boundary, and the exit boundaries are specified as Dirichlet boundary conditions. A modified successive over-relaxation (SOR) technique is used to solve equations (E.5) and (E.6), thereby locating the interior points. Like a normal SOR procedure, the iterative corrections at each point are over-relaxed. However, corrections to the values of the nonhomogeneous terms, P and Q , are under-relaxed each iteration. A detailed description of these nonhomogeneous terms, as they are handled in this investigation, is presented in the following paragraphs.

E.2.1 CONTROL OF GRID GEOMETRY NEAR BOUNDARIES. For the grids generated to support this investigation, the nonhomogeneous terms in the Poisson equations are used to control grid geometry based on the features of the grid at the $\eta = 1$ and $\eta = \eta_{\max}$ boundaries. Along the blade surface, grid

cut, and inflow boundaries, the following two features are controlled:

1. Grid line spacings from points on the inner and outer boundaries to the corresponding points on the first grid line inside each of these boundaries are specified.
2. Intersection angles between these same boundaries and the $\xi = \text{constant}$ grid lines intersecting them are specified.

Similarly, along the periodic boundaries the following two features are controlled:

1. Grid line spacings from points on the inner and outer boundaries to the corresponding points on the first grid line inside each of these boundaries are specified.
2. Slope of the $\xi = \text{constant}$ grid lines in the physical plane where they cross the periodic boundaries is specified. The slope used is the tangent of the cascade stagger angle.

Figures E-5 and E-6 illustrate these effects. Figures E-5 and E-6 each consist of two identical grids joined at the periodic boundary. Figure E-5 presents a Laplace-type grid while Figure E-6 presents a Poisson-type grid with the control features described above. Notice that, in addition to controlling geometric features near the boundaries, the Poisson source terms also control grid point distribution in the interior of the grid.

The Laplace-type grid has a very large range of grid cell sizes. The very large grid cells may cause prohibitively large flow solver truncation errors in

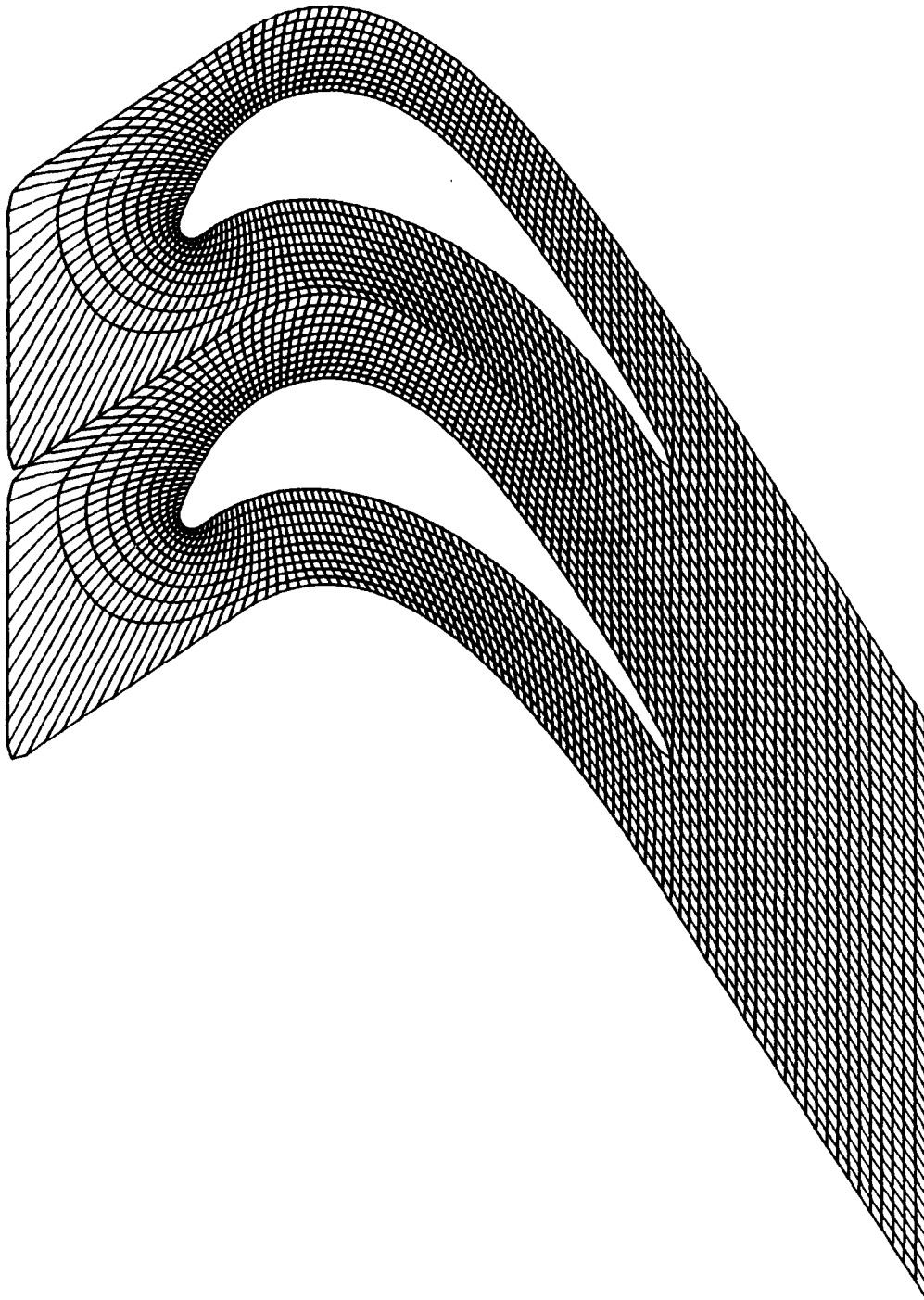


Figure E-5. C-type grid generated by the Laplace equation.

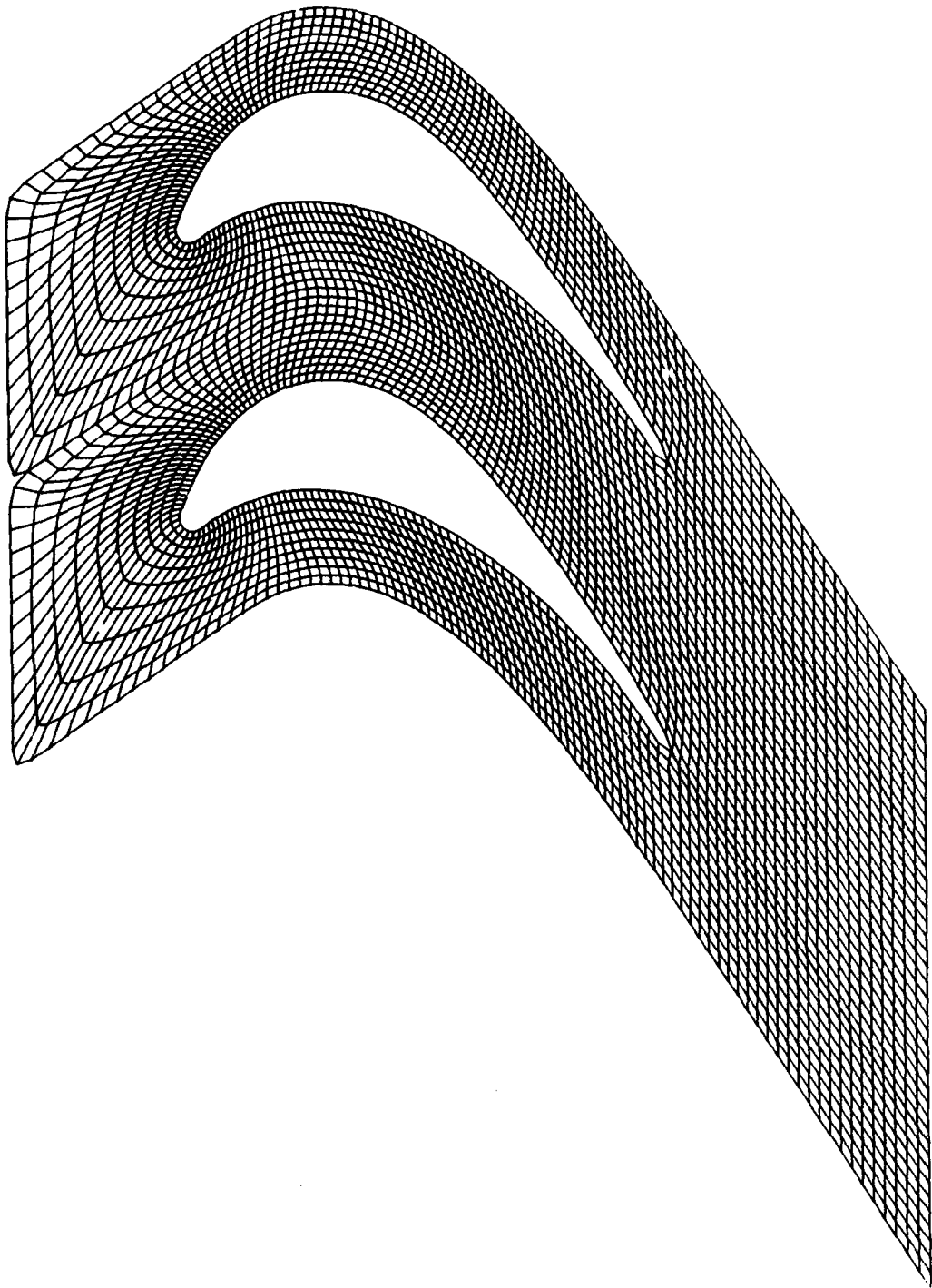


Figure E-6. C-type grid generated by the Poisson equation.

these regions of the grid. In addition, there is insufficient control of the skewness of grid line intersections. Excessive skewness increases truncation errors and therefore, should be minimized.

Accuracy is also reduced by rapid changes in grid cell sizes and by discontinuities in the derivatives of the grid lines (i.e., grid line slope, curvature, etc.). In the limit of infinitely small grid cells, Laplace and Poisson equations each insure completely smooth and differentiable grid lines in the grid interior. However, with a periodic boundary like the one joining the grids in Figures E-5 and E-6, control must be exercised over the grid features at the periodic boundary to avoid sharp discontinuities in the grid line slopes and rapid changes in the grid cell sizes across these boundaries. Each of the problems described above is evident in the Laplace-type grid illustrated in Figure E-5. However, all of these problems have been overcome in the Poisson-type grid illustrated in Figure E-6.

E.2.1.1 GRID CONTROL AT INFLOW, BLADE SURFACE, AND GRID CUT BOUNDARIES. Steger and Sorenson [1] have developed the techniques required to implement the controls described above. Some modifications to their method have been developed for this investigation. Steger and Sorenson start by solving equations (E.5) and (E.6) for P_b and Q_b , which are the P and Q values on the boundaries.

$$P_b = J(y_\eta R_1 - x_\eta R_2) \Big|_{\eta_{\text{boundary}}} \quad (\text{E.7})$$

$$Q_b = J(-y_\xi R_1 + x_\xi R_2) \Big|_{\eta_{\text{boundary}}} \quad (\text{E.8})$$

where

$$R_1 = -J^2(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}) \Big|_{\eta_{\text{boundary}}}$$

$$R_2 = -J^2(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}) \Big|_{\eta_{\text{boundary}}}$$

Values of P_b and Q_b are calculated at each point on each of the $\eta = \text{constant}$ boundaries using equations (E.7) and (E.8). Since the physical locations of all points on these boundaries are specified, x_ξ , y_ξ , $x_{\xi\xi}$, and $y_{\xi\xi}$ are calculated directly from the specified boundary point locations.

The key to Steger and Sorenson's method is their evaluation of x_η and y_η . They derive expressions for x_η and y_η by first considering the offset distance, ΔS , which is the distance in the physical space from the boundary point, along a $\xi = \text{constant}$ grid line, to the corresponding point on a $\eta = \text{constant}$ grid line inside the boundary. Thus,

$$\Delta S = [(\Delta x)^2 + (\Delta y)^2]^{1/2} \Big|_{\xi=\text{constant}} \quad (\text{E.9})$$

In the limit:

$$dS = [(dx)^2 + (dy)^2]^{1/2} \Big|_{\xi=\text{constant}} \quad (\text{E.10})$$

From the partial differentiation chain rule:

$$dS = [(x_\xi d\xi + x_\eta d\eta)^2 + (y_\xi d\xi + y_\eta d\eta)^2]^{1/2} \Big|_{\xi=\text{constant}} \quad (\text{E.11})$$

Since $dS/d\eta$ represents the distance between two points on the same $\xi = \text{constant}$ grid line, $d\xi = 0$. Therefore, equation (E.11) reduces to:

$$\frac{dS}{d\eta} = [(x_\eta)^2 + (y_\eta)^2]^{1/2} \Big|_{\xi=\text{constant}} \quad (\text{E.12})$$

The second grid geometry control feature which Steger and Sorensen address is the grid line intersection angle at the periodic boundary. This control is implemented using the vector dot product:

$$\nabla\xi \cdot \nabla\eta = |\nabla\xi| |\nabla\eta| \cos\theta \quad (\text{E.13})$$

where θ is the desired intersection angle ($\theta = 90^\circ$ yields orthogonal grid line intersections). Equation (E.13) is the definition of the dot product of a vector normal to the $\xi = \text{constant}$ grid line with a vector normal to the $\eta = \text{constant}$ grid line. Carrying out the vector arithmetic and incorporating the two-dimensional transformation metrics ($\xi_x = J y_\eta$, $\xi_y = -J x_\eta$, $\eta_x = -J y_\xi$, and $\eta_y = J x_\xi$) yields:

$$x_\xi x_\eta + y_\xi y_\eta = -[(x_\eta^2 + y_\eta^2)(x_\xi^2 + y_\xi^2)]^{1/2} \cos\theta \quad (\text{E.14})$$

Solving equations (E.12) and (E.14) simultaneously to determine the necessary values of x_η and y_η required to meet these two conditions yields:

$$x_\eta = \frac{dS}{d\eta} \frac{(-x_\xi \cos\theta - y_\xi \sin\theta)}{(x_\xi^2 + y_\xi^2)^{1/2}} \quad (\text{E.15})$$

$$y_{\eta} = \frac{dS}{d\eta} \frac{(-y_{\xi} \cos \theta + x_{\xi} \sin \theta)}{(x_{\xi}^2 + y_{\xi}^2)^{1/2}} \quad (\text{E.16})$$

Both θ and $dS/d\eta$ are specified at every ξ location on each of the $\eta = \text{constant}$ boundaries.

Once the values of x_{η} and y_{η} on the boundaries are known, the cross derivatives $x_{\xi\eta}$ and $y_{\xi\eta}$ needed in equations (E.7) and (E.8) are calculated numerically by differencing x_{η} and y_{η} with respect to ξ . Therefore, the only remaining unknowns on the right-hand side of equations (E.7) and (E.8) are $x_{\eta\eta}$ and $y_{\eta\eta}$. The approach used by Steger and Sorenson is to numerically calculate $x_{\eta\eta}$ and $y_{\eta\eta}$ during each SOR iteration using the one-sided approximations shown in equations (E.17) and (E.18) for the $\eta = 1$ boundary:

$$x_{\eta\eta} = \frac{-7x_1 + 8x_2 - x_3}{2(\Delta\eta)^2} - \frac{3x_{\eta}}{\Delta\eta} \bigg|_{\eta=1} \quad (\text{E.17})$$

$$y_{\eta\eta} = \frac{-7y_1 + 8y_2 - y_3}{2(\Delta\eta)^2} - \frac{3y_{\eta}}{\Delta\eta} \bigg|_{\eta=1} \quad (\text{E.18})$$

and the one-sided approximations shown in equations (E.19) and (E.20) for the η_{\max} boundary:

$$x_{\eta\eta} = \frac{-7x_{\eta_{\max}} + 8x_{(\eta_{\max}-1)} - x_{(\eta_{\max}-2)}}{2(\Delta\eta)^2} + \frac{3x_{\eta}}{\Delta\eta} \bigg|_{\eta_{\max}} \quad (\text{E.19})$$

$$y_{\eta\eta} = \frac{-7y_{\eta_{\max}} + 8y_{(\eta_{\max}-1)} - y_{(\eta_{\max}-2)}}{2(\Delta\eta)^2} + \frac{3y_{\eta}}{\Delta\eta} \bigg|_{\eta_{\max}} \quad (\text{E.20})$$

Once all the necessary first and second derivatives have been determined, P_b and Q_b are calculated using equations (E.7) and (E.8). Thus, most

derivatives needed to compute P_b and Q_b can be calculated once and stored, but $x_{\eta\eta}$ and $y_{\eta\eta}$ must be recalculated every iteration prior to calculating P_b and Q_b . Corrections to P_b and Q_b are severely under-relaxed each SOR iteration. Their under-relaxation coefficients are increased from initial values of approximately 0.01, to approximately 0.05 as convergence is approached.

In order to smoothly propagate the P and Q effects throughout the grid, these quantities are exponentially interpolated into the interior of the grid using equations (E.21) and (E.22):

$$P(\xi, \eta) = P(\xi, 1) e^{-a(\eta-1)/(\eta_{\max}-1)} + P(\xi, \eta_{\max}) e^{-b(\eta_{\max}-\eta)/(\eta_{\max}-1)} \quad (E.21)$$

$$Q(\xi, \eta) = Q(\xi, 1) e^{-c(\eta-1)/(\eta_{\max}-1)} + Q(\xi, \eta_{\max}) e^{-d(\eta_{\max}-\eta)/(\eta_{\max}-1)} \quad (E.22)$$

The terms $a = a(\xi)$, $b = b(\xi)$, $c = c(\xi)$, and $d = d(\xi)$ are positive decay rates used to control the exponential decay of the P and Q effects into the interior of the grid. Equations (E.21) and (E.22) differ slightly from those developed by Steger and Sorenson in that the exponents have been normalized by $(\eta_{\max}-1)$. This normalization allows the user to change grid density (i.e., the number of $\eta = \text{constant}$ lines) without having to change the decay rates proportionately. A zero finding secant method is used to home in on the appropriate decay rates at each grid point on each of the $\eta = \text{constant}$ boundaries.

Since the theory behind this technique is developed in the limit of infinitely small space increments, but is applied to a finite size grid, the exact grid control requested will be approached, but not achieved exactly.

E.2.1.2 GRID CONTROL AT PERIODIC BOUNDARIES. A variation of Steger and Sorenson's technique has been developed during this investigation for the periodic boundaries of the cascade grid. This new technique is described in this section. Steger and Sorenson's technique allows grid line intersection angles to be specified, without regard to the slopes of those grid lines in the physical space. Often, this is exactly the control desired. However, sometimes the actual slope of a grid line in the physical space must be specified, with only cautious monitoring of grid line intersection angles. The periodic boundaries on a C-type cascade grid are an excellent example of grid line slope specification being superior to grid line intersection angle specification. Figure E-7 shows graphically that when grid lines pass through the periodic boundary at other than the stager angle, γ , unnecessary skewness results in the interior of the grid. Therefore, forcing the $\xi = \text{constant}$ grid lines to intersect the periodic boundaries at the cascade stager angle will both minimize skewness in the interior of the grid and produce continuous grid line slopes at the periodic boundary.

This type of slope control can be achieved by Steger and Sorenson's technique. However, it is unnecessarily cumbersome just calculating the intersection angle, θ , that corresponds to the stager angle, γ . A much more efficient method has been developed to achieve the same results. If grid line spacing and slope control is desired along an $\eta = \text{constant}$ boundary, the following equations replace equations (E.15) and (E.16):

$$x_{\eta} = \frac{dS}{d\eta} \sin(\gamma) \quad (\text{E.23})$$

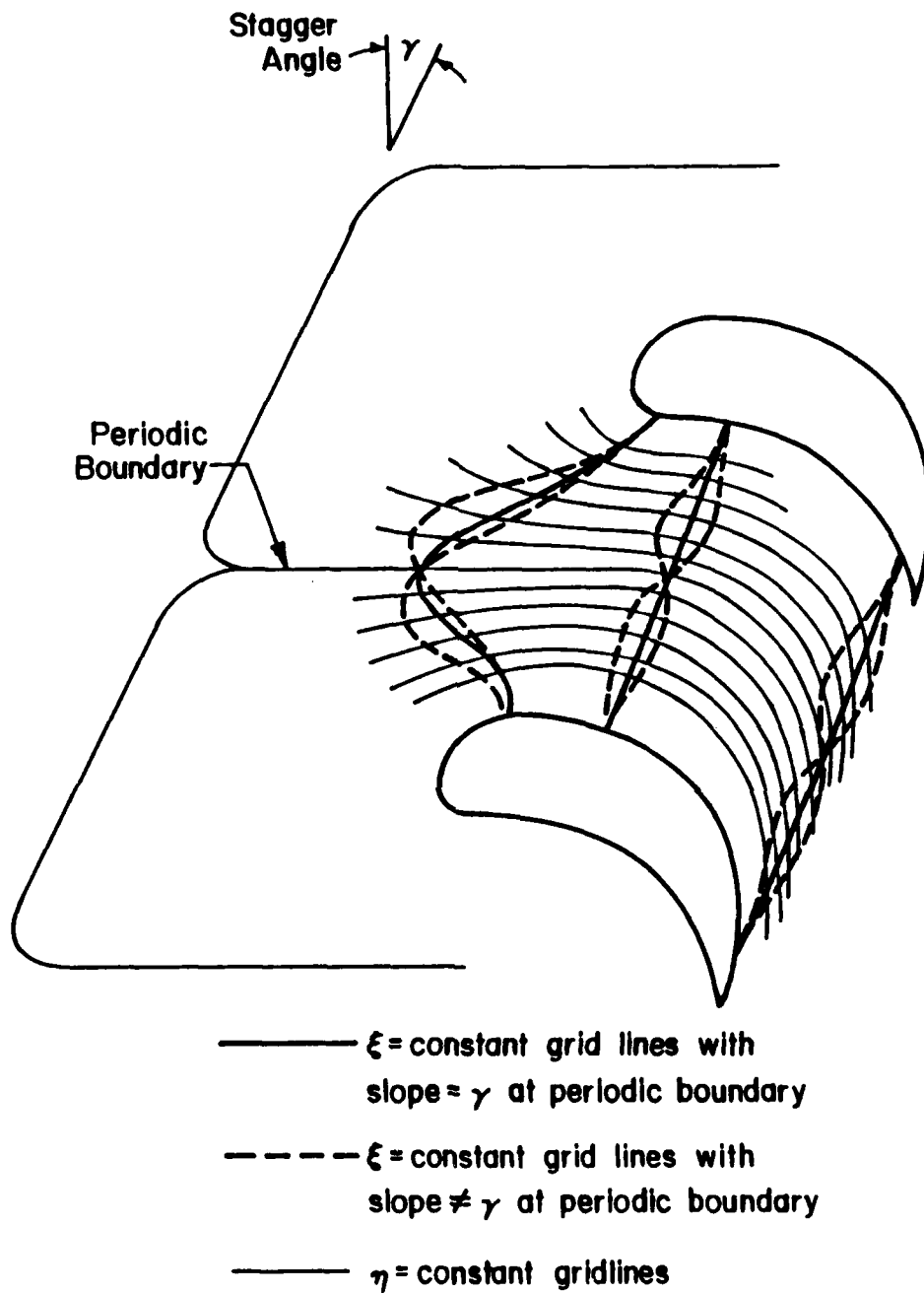


Figure E-7. Slope control of grid lines crossing the periodic boundaries.

$$y_{\eta} = \frac{dS}{d\eta} \cos(\gamma_i) \quad (E.24)$$

In the special case of zero stagger angle (i.e., vertical $\xi = \text{constant}$ grid lines at the periodic boundary), equations (E.23) and (E.24) reduce to:

$$x_{\eta} = 0 \quad (E.25)$$

$$y_{\eta} = \frac{dS}{d\eta} \quad (E.26)$$

This technique of specifying the slopes of the intersecting grid lines, rather than the intersection angles, is employed at the periodic boundaries in the present investigation.

E.3 GRID POINT DISTRIBUTION ON BOUNDARIES

The placement of grid points on the boundaries of the physical grid is as important as the grid interior control features discussed above. This section discusses several considerations concerning boundary grid point distribution and outlines how these considerations are incorporated in the present investigation.

The following factors should be considered to properly distribute points around the boundaries of a cascade blade, C-type grid.

1. Points should be placed in a manner which will provide an appropriate balance between flow feature resolution and computational efficiency. In regions of the flow where high property gradients exist (stagnation points, shock waves, flow velocities near Mach one), the grid points

should be concentrated to resolve these flow features. On the other hand, for the sake of computational efficiency, grid points should be spread out in regions of low property gradients.

2. Points should be concentrated in regions where fine refinement of physical geometry is required, or conversely, points may need to be sparse in regions where grid refinement could cause flow solver instabilities.
3. Smooth transitions in grid point density should be incorporated inbetween regions of high and low point density.
4. The relative position of corresponding points on the inner and outer boundary boundaries should minimizes skewness between the $\xi = \text{constant}$ and the $\eta = \text{constant}$ grid lines throughout the grid.
5. Blade geometry is normally provided as a series of discrete (x,y) values. Automatic curve fitting must be incorporated which will appropriately distribute grid points on the blade surface. These new (x,y) locations may not necessarily be the same in number or location as the input values.
6. Overall grid shape must smoothly and appropriately incorporate cascade stagger angle, blade spacing, blade camber, and blade thickness.
7. The location of each point on the upper side of the grid cut (trailing edge to exit) must be exactly the same as the location of the corresponding point on the lower side of the grid cut.

8. The location of each point on the periodic boundary on the upper side of the grid must differ from the corresponding lower side point location only by the appropriate cascade blade spacing and stagger angle offsets.

E.3.1 BOUNDARY POINT PLACEMENT SEQUENCE. The placement of points around the boundary of a cascade blade grid is handled in two basic steps. First, the boundaries are broken into several small segments and each of these segments is assigned a location in physical space. Second, grid points are distributed appropriately on each segment.

In the present investigation, the cascade grid boundaries are broken into the 15 segments shown on Figure E-8. Each of these segments is assigned a location in physical space as a function of cascade geometry in accordance with the guidelines described above. The three segments which lie on the blade surface are constructed using a cubic spline curve fit routine developed by Akima [20].

Once locations are specified for each of the boundary segments, point locations on each segment are determined using polynomial curve fits of first, second, and third order. Each of these polynomials distribute points along the arc length of the segment as a function of ξ or η . Of course, grid point locations match at each of the segment intersections. In addition, the grid point spacing is matched at each of the segment intersections.

Once boundary point locations are determined, this information is passed to the Poisson grid solver where interior point locations are determined.

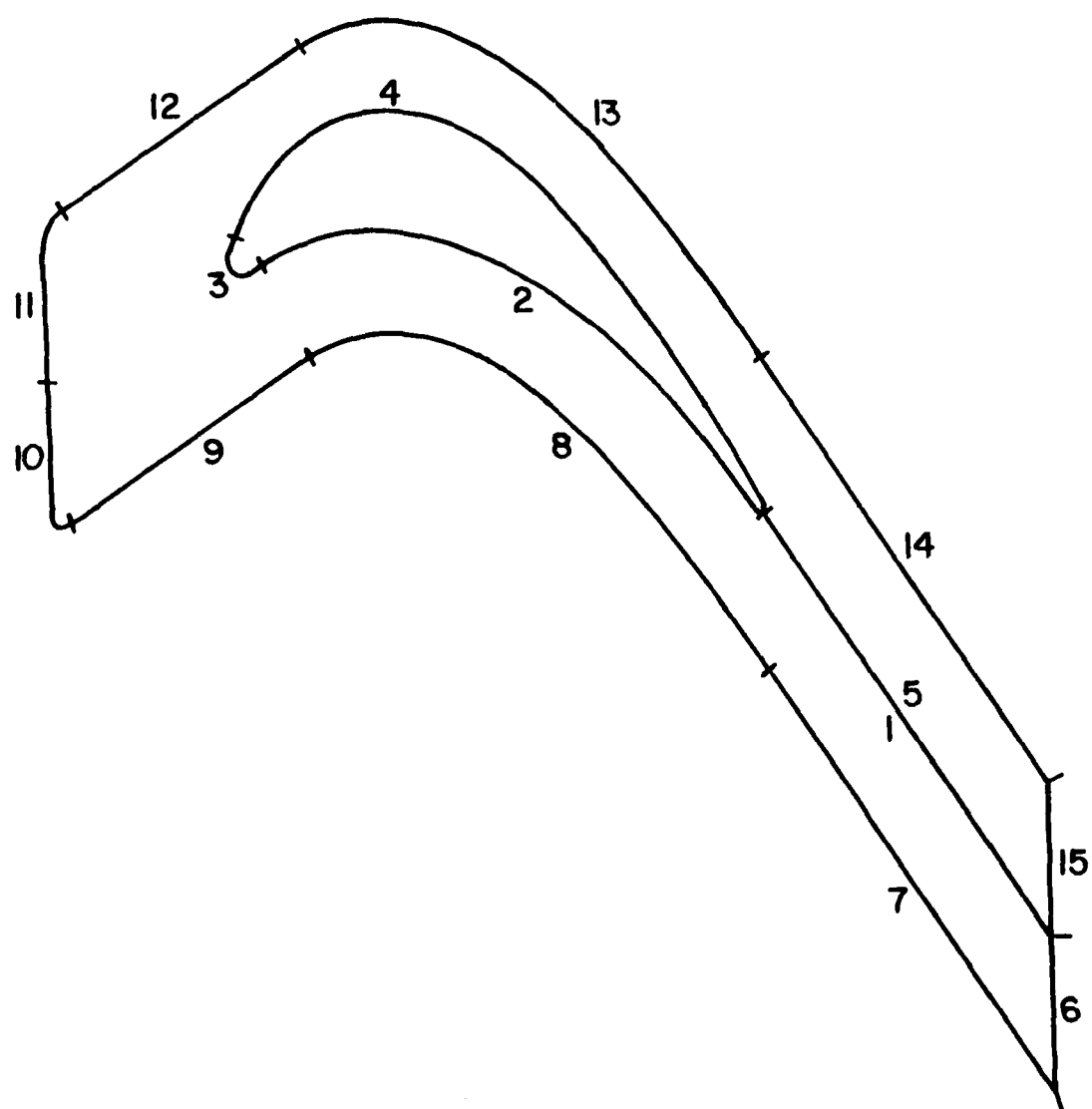


Figure E-8. Grid boundaries broken into 15 segments.

E.4 ADDITIONAL GRID GENERATION CAPABILITIES

The cascade grids used in the present investigation have a zero stagger angle and a constant blade spacing. The grid generator described in this appendix is also capable of creating grids which have nonzero stagger and systems of grids around detuned cascades. This section contains examples of such grids. In each figure an extra grid has been drawn to illustrate the flow channel and the cascade periodicity.

Figure E-9 illustrates a C-type grid with a nonzero stagger angle.

The three grids presented in Figure E-10 illustrate a detuned cascade. Unlike conventional cascade studies, flow solutions on this detuned cascade require a system of two C-grids since the cascade repeats itself after every other blade. These grids are being used by Chiang and Fleeter [21] to study unsteady flows through detuned cascades.

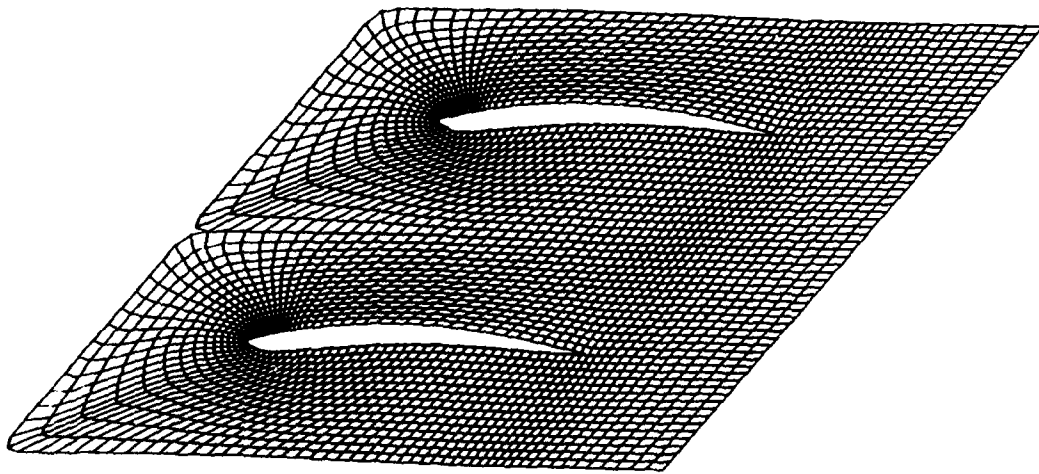


Figure E-9. Cascade with nonzero stagger.

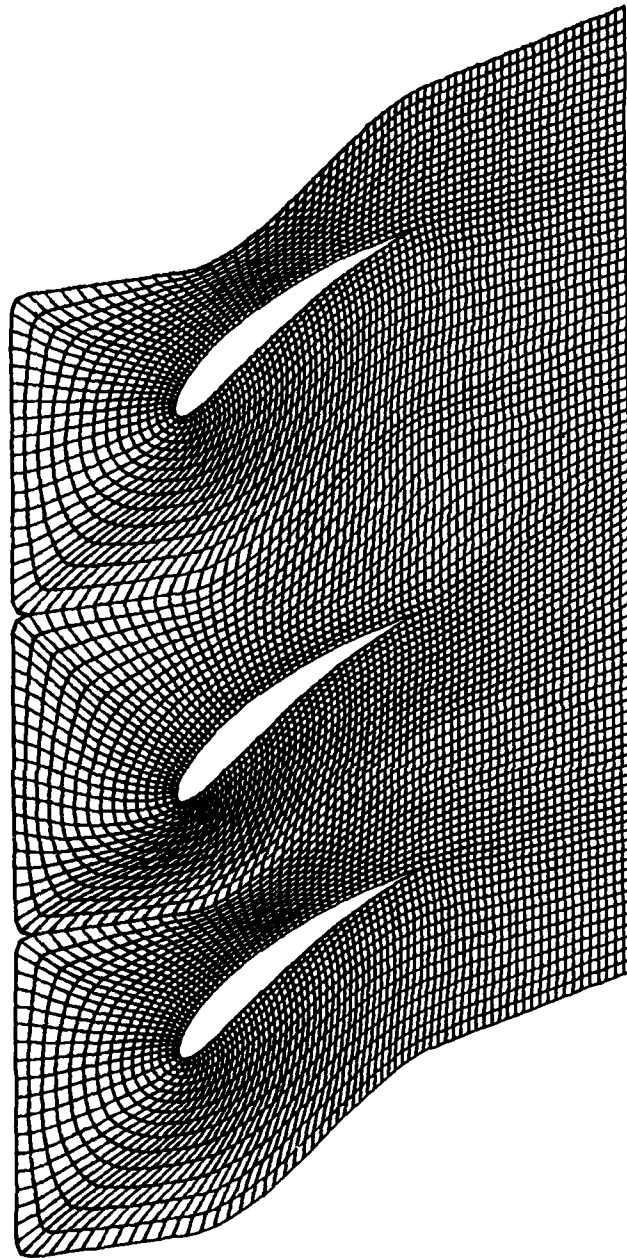


Figure E-10. Grid system for a detuned cascade.

APPENDIX F

EXPLICIT ARTIFICIAL DISSIPATION

A numerical solution to the unsteady Euler or Navier-Stokes equations can be limited by stability considerations. In other words, if the numerical solution to a set of unsteady partial differential equations diverges rather than predicting a bounded solution of the equations, no useful information is derived. A stability analysis for a particular numerical method applied to a particular set of partial differential equations produces a time step limitation which must be imposed by the analyst to avoid divergence. Unfortunately, common stability analysis techniques predict the stability criterion applicable to simplified linear model equations in the absence of the boundaries of the domain. Therefore, even when the calculated stability restrictions are enforced, instabilities can still occur. In particular, when numerically solving the Euler or Navier-Stokes equations, nonlinearities in the equation formulations, various boundary condition implementations, and flow solution phenomena such as shock waves and stagnation points, often cause numerical instabilities. Explicit artificial dissipation, or smoothing, is usually applied to overcome these instabilities.

Numerical dissipation involves smoothing out the locally high and low property values across the solution domain. In other words, if the value of density at a grid point is higher than the values of density at the surrounding grid points, dissipation would lower the density at that point and raise the

surrounding values. The effect is to damp local instabilities. Unfortunately, dissipation can also cause the flow solution to be smeared to the point that resolution of flow features is lost.

The term artificial dissipation describes numerical dissipation which does not model any real physical phenomena. The Euler equations do not contain any dissipative terms. The viscous terms in the Navier-Stokes equations are dissipative. However, the level of real dissipation in a viscous flow may not be sufficient to damp numerical instabilities.

When partial derivatives are modeled numerically, the result is not completely accurate. The numerical approximation actually models the derivative plus some unwanted higher-order terms. These higher order terms are often ignored. However, sometimes these terms produce artificial dissipation effects. Since this artificial dissipation is a natural result of using the numerical method, it is referred to as implicit artificial dissipation.

When the modeled physical dissipation (if any) and any implicit artificial dissipation resulting from inexact derivative approximations are insufficient to damp instabilities, additional artificial dissipation is explicitly added. This dissipation is referred to as explicit artificial dissipation.

The following are desirable characteristics of explicit artificial dissipation.

1. Explicit artificial dissipation should cause the numerical model to produce a stable solution.
2. Explicit artificial dissipation should not cause the numerical solution to differ significantly from the real physical solution. In other words, the

dissipation should not unnecessarily smear flow features.

3. Explicit artificial dissipation should be computationally efficient.

F.1 EXPLICIT ARTIFICIAL DISSIPATION MODELS

Since explicit artificial dissipation does not model any real physical phenomena, there are many acceptable approaches to its implementation. However, a review of the literature reveals some commonalities between techniques. For example, numerical dissipation is caused by even-order derivative approximations. Therefore, explicit artificial dissipation methods are usually based on some combination of second- and fourth-order derivatives of the properties being smoothed. The following general equation is representative of many explicit artificial dissipation models:

$$f_{\text{smoothed}} = f + \Delta t \left[c_2 \left(pc\xi_2 sf\xi_2 f_{\xi\xi} + pc\eta_2 sf\eta_2 f_{\eta\eta} \right) - c_4 \left(pc\xi_4 sf\xi_4 f_{\xi\xi\xi\xi} + pc\eta_4 sf\eta_4 f_{\eta\eta\eta\eta} \right) \right] \quad (F.1)$$

where f represents any fluid property being smoothed, Δt is the time step at the point where the smoothing is taking place, all the pc terms are property dependent coefficients, c_2 and c_4 are property independent coefficients, and all the sf terms are grid dependent scaling factors. Each component of equation (F.1) is discussed below.

Consideration of the numerical approximations of second and fourth derivatives illustrates why they are dissipative. Equation (F.2) is the standard

centered second-order approximation of a second derivative with respect to ξ :

$$F_{\xi\xi} = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta\xi^2} \quad (\text{F.2})$$

Equation (F.3) is the standard centered second-order approximation of a fourth derivative with respect to ξ :

$$F_{\xi\xi\xi\xi} = \frac{f_{i+2} - 4f_{i+1} + 6f_i - 4f_{i-1} + f_{i-2}}{\Delta\xi^4} \quad (\text{F.3})$$

As with any numerical approximation of a derivative, the sum of all the coefficients in each of these approximations is zero. In equation (F.2) the property value at the central point is subtracted while the values at the surrounding points are added. If this derivative approximation, times a fractional coefficient, is added to the value of the property at the central point, f_i , it makes the result a weighted average of the value at the point in question and the surrounding values. If the point in question had a relatively high value, it would be lowered, and vice versa.

Similarly for the fourth derivative. If the fourth derivative approximation times a fractional coefficient is subtracted from the property value at the central point, f_i , it produces a weighted average of the property value at i and four neighboring points.

The second and fourth derivative terms in equation (F.1) are not used interchangeably. The second derivative terms are used to damp large and small instability problems. Fourth derivative terms are used to damp small oscillations over the entire flowfield. Due to the five point stencil

characteristic of second-order accurate numerical methods applied to two-dimensional physical spaces, a numerical decoupling of the odd and even grid lines can occur in the flow solution. Fourth-order dissipation does an excellent job of smoothing the property value oscillations that are caused by this decoupling. On the other hand, as Pulliam [22] demonstrates, due to its large nine point stencil, fourth derivative smoothing creates oscillations across strong flowfield features such as shock waves.

All the smoothing terms in equation (F.1) are multiplied by the time step. This causes the smoothing to be scaled at the same magnitude as the space derivative terms in the Euler or Navier-Stokes equations. In addition, if local time steps are being used rather than global time steps, multiplication by the time step will cause more explicit artificial dissipation to be added where larger time steps may be allowing instabilities to grow faster.

The second-order property dependent coefficients, $pc\xi_2$ and $pc\eta_2$, produce two effects. They cause the artificial dissipation to resemble viscous dissipation. They also can be used to automatically concentrate large amounts of dissipation in areas where instability is most threatening.

In a viscous flow, the greatest effects of viscosity are felt where the second derivatives of velocity are the highest. Similarly, when $pc\xi_2$ and $pc\eta_2$ are directly proportional to the second derivative of a property, the artificial dissipation is concentrated in regions where the second derivatives of the scaling property are large. Typically, pressure or density is used as the scaling property in primitive variable or conservative variable flow solvers,

respectively. The scaling property derivatives are usually normalized by the magnitude of the scaling property. For example a common scaling factor is the second derivative of pressure divided by pressure.

In order to spread out and more smoothly apply the effects of explicit artificial dissipation, a property dependent scaling factor can be made proportional to the largest second derivative of the scaling property on any grid point in the five point stencil associated with the grid point being smoothed. As these schemes become more elaborate, they also tend to become more computationally demanding.

By scaling explicit artificial dissipation with property derivatives, the dissipative effects are automatically concentrated in areas where instability is likely to occur or is already evident. Some flowfield regions, such as near shock waves or near airfoil trailing edges, are characterized by rapid changes or discontinuities in property values. These rapid changes or discontinuities often cause numerical instability. In the early stages of development, numerical instability is usually evidenced by roughness in solution property value surfaces. Therefore, in unstable regions of the flowfield and near physical phenomena where instability is likely, the second derivatives of pressure and density are large. Thus, scaling in proportion to these second derivatives applies the dissipation where it is most needed.

Multiplying $pc\xi_2$ or $pc\eta_2$ times their respective dissipative second derivatives produces the product of two second derivatives. By convention this product is sometimes referred to as product fourth-order dissipation. This

should not to be confused with the actual fourth-order dissipation terms in equation (F.1).

Fourth derivative dissipation usually requires more computational effort than second derivative dissipation. Also, on the relatively coarse grids used for inviscid calculations, the nine point stencil used to calculate fourth derivatives can stretch across a large region of the grid. This can cause excessive smearing. In addition, fourth derivative dissipation is not normally required for stability. Therefore, unless a very high level of convergence is required or aesthetic smoothing of small wiggles in the property values is desired, the fourth derivative terms can be eliminated completely.

Like $pc\xi_2$ and $pc\eta_2$, the fourth derivative scaling factors, $pc\xi_4$ and $pc\eta_4$, are property dependent. However, rather than using roughness in property value surfaces to increase these dissipation coefficients, $pc\xi_4$ and $pc\eta_4$ are normally decreased in regions where property value surfaces are rough. As discussed earlier, fourth-order dissipation causes oscillations across flow features such as shock waves. Therefore, in these regions of the flowfield, researchers such as Jameson [23] and Pulliam [22] gain stability through second-order explicit artificial dissipation and force the fourth derivative coefficients to zero.

Calculating $pc\xi_2$, $pc\eta_2$, $pc\xi_4$, and $pc\eta_4$ requires a significant amount of computational effort. In many flowfields these scaling factors are not needed. Therefore, these factors are often all set to a constant value of 1.0. Since this removes the fluid property dependence from the coefficients of the damping terms, much of the similarity with natural dissipation is lost. The resulting

simpler damping terms are often referred to as smoothing rather than artificial dissipation.

The property independent coefficients, c_2 and c_4 , are used whether or not property dependent coefficients are used. These coefficients set the magnitude of the dissipation over the entire flowfield. In some cases it is beneficial to vary these coefficients as a function of time. For example, c_2 might be linearly decreased from a value of 1.0 to a value of 0.4 over the period of 1000 time steps. This may keep a rough initial-value surface from making the solution diverge during the first few time steps while still minimizing flow feature smearing of the converged solution.

The coefficients $sf\xi_2$, $sf\eta_2$, $sf\xi_4$, and $sf\eta_4$ scale the dissipative terms to correct unwanted, grid dependent scaling of the dissipative derivatives. The dissipative derivatives in equation (F.1) are all taken with respect to the computational variables (ξ, η) . On the computational grid, all grid lines are equally spaced. On the physical grid, the grid lines are not equally spaced. Therefore, derivatives with a given magnitude in physical space will, when differentiated in computational space, have a smaller magnitude in densely packed regions of the grid than in sparsely packed regions of the grid. Without the sf correction factor terms, sufficient smoothing near densely packed areas of the grid will cause excessive smoothing in relatively coarse areas of the grid. To accentuate this problem still further, grids are often packed near boundaries where many instabilities originate. Fortunately, if the grids are stationary, the scaling factor terms can be calculated once, at the beginning of the flow study, stored, and used at each time step.

F.2 EXPLICIT ARTIFICIAL DISSIPATION NEAR BOUNDARIES

The boundaries of the computational domain can be the source of some numerical instabilities. However, special consideration must be given to implementing explicit artificial dissipation near flowfield boundaries. In fact, the subtle problems encountered when smoothing is applied perpendicular to boundaries are so difficult that it is usually considered best not to smooth perpendicular to boundaries. The following paragraphs explain why simple extensions of interior point dissipation implementations can produce unacceptable results.

Fluid properties are computed only on one side of the grid points which lie on the computational boundaries of a flowfield. Therefore, equations (F.2) and (F.3) can not be applied at these points, perpendicular to the boundary, without some extra provision. Several possible modifications to the interior point implementations exist. Four possibilities for smoothing at the boundaries are listed and discussed below. At first these options appear to make smoothing at the boundaries simple and feasible. However, closer examination reveals that none of these options are profitable.

1. Use linear extrapolation to approximate property values outside the boundary. Then use a standard, second-order derivative approximations at the boundary point to smooth.
2. Use quadratic extrapolation to approximate property values outside the boundary. Then use a standard, second-order derivative approximations at the boundary point to smooth.

3. Use the standard one-sided second derivative approximation at the boundary point to smooth.
4. Use the standard one-sided first derivative approximation at the boundary point to smooth.

The standard linear extrapolation formula is:

$$f_{0,\eta} = 2f_{1,\eta} - f_{2,\eta} \quad (\text{F.4})$$

where, for this example, the boundary point is the $\xi=1$ grid point. Substituting equation (F.4) into equation (F.2) with $i=1$ yields exactly zero. This is to be expected since a linear extrapolation approximates a straight line and therefore, no second derivatives can exist.

The standard quadratic extrapolation formula is:

$$f_{0,\eta} = 3f_{1,\eta} - 3f_{2,\eta} + f_{3,\eta} \quad (\text{F.5})$$

where, again, the boundary point is the $\xi=1$ grid point. Substituting equation (F.5) into equation (F.2) yields the following approximation for the second derivative of f with respect to ξ at a $\xi=1$ boundary:

$$f_{\xi\xi} = f_{1,\eta} - 2f_{2,\eta} + f_{3,\eta} \quad (\text{F.6})$$

Unfortunately, using equation (F.6) to smooth at point $(1,\eta)$ would have the opposite effect. If this derivative approximation, times a fractional coefficient, is added to $f_{1,\eta}$, the difference between $f_{1,\eta}$ and $f_{2,\eta}$ would increase, rather than decrease! Thus, instead of damping instabilities, they are excited.

Instead of extrapolating across the boundary and using centered differences to approximate derivatives, one-sided approximations might be used. The

one-sided second derivative approximation at point $(1,\eta)$ is exactly equation (F.6). Therefore, this approach would also be counter productive.

The one-sided approximation of the first derivative of f with respect to ξ at point $(1,\eta)$ is:

$$f_{\xi} = \frac{f_{2,\eta} - f_{1,\eta}}{\Delta\xi} \quad (\text{F.7})$$

If this derivative approximation, times a fractional coefficient, is added to $f_{1,\eta}$, the difference between $f_{1,\eta}$ and $f_{2,\eta}$ would be decreased, as desired. However, this first derivative is convective, rather than dissipative. Therefore, rather than averaging surrounding points, the effect would be to convect interior point property values to the boundaries and significantly alter the flowfield solution.

Other possibilities exist for smoothing perpendicular to boundaries. However, care must be exercised to avoid variations of the same problems encountered above. One usable scheme is to use the one-sided second derivative approach, equation (F.6), described in (3) above, but multiply the smoothing term by minus 1.0.

Because of the difficulties associated with implementing explicit artificial dissipation perpendicular to computational boundaries, many researchers do not smooth perpendicular to boundaries. Instead, enough explicit dissipation is added throughout the remainder of the grid to control instabilities.

Until now no mention has been made of implementing explicit artificial dissipation along (parallel to) computational boundaries. Smoothing along

computational boundaries is feasible. However, in some cases special precautions must still be taken. For example, along a free slip, impermeable boundary, when the velocity is smoothed, extra effort should be applied to insure the resulting velocity vector is tangent to the boundary.

F.3 EXPLICIT ARTIFICIAL DISSIPATION FOR THE PRESENT INVESTIGATION

In the first two sections of this appendix, a variety of explicit artificial dissipation techniques and special considerations are discussed. Obviously, explicit artificial dissipation schemes can be simple or complex. They can also be efficient or very demanding computationally. For the sake of economy, explicit artificial dissipation should be kept as simple as possible. This section describes the approach to explicit artificial dissipation used in the present investigation.

The current research effort employs a conservation variable form of the Kentzer method [2] at the flowfield boundaries and the MacCormack method [3] throughout the flowfield. This combination requires some explicit artificial dissipation to avoid instabilities. Specifically, the blade surface boundary appears to generate instabilities. However, when a small amount of smoothing is applied over the majority of the grid, excellent results are achieved.

For the sake of computational efficiency, only second derivative smoothing is applied. In other words, the fourth derivative terms are not used, and $pc\xi_2$ and $pc\eta_2$ are each set to 1.0. Therefore, the general formula used to smooth each of the four conservation variable properties ρ , ρu , ρv , and ρe , is:

$$f_{\text{smoothed}} = f + \Delta t c_2 \left[sf\xi_2 f_{\xi\xi} + sf\eta_2 f_{\eta\eta} \right] \quad (\text{F.8})$$

At the blade surface and trailing edge point boundaries, smoothing is only applied tangent to the boundaries. At these boundaries the momentum magnitude is smoothed and then split into x and y components to insure tangency.

No smoothing is applied at the inflow or the exit boundaries.

Interior points on periodic boundaries and on grid cut boundaries are treated like other interior points but with special indexing required due to the boundary locations on the computational grid.

The property independent coefficient, c_2 , is linearly decreased in time from an initial-value to approximately half that value in the following manner. The upper and lower limits of c_2 are set along with a multiplying factor. At each time step, the value of c_2 is multiplied by the multiplying factor until the lower limit of c_2 is reached. At all subsequent time steps c_2 is maintained at the lower limit.

As described in Section F.1 of this appendix, the grid dependent scaling factors, $sf\xi_2$ and $sf\eta_2$, are used to correct unwanted, grid dependent scaling of the dissipation derivatives, $f_{\xi\xi}$ and $f_{\eta\eta}$. Without these factors, the dissipation derivatives have less effect in areas of the physical grid where grid points are densely packed than in areas where grid points are sparse. To eliminate this effect, a procedure has been developed to rescale property derivatives which are taken with respect to computational directions.

Consider a curve in physical space, s , which follows any computational grid line. For illustrative purposes, assume s follows an $\eta = \text{constant}$ grid line. Differentiating the dissipative derivatives with respect to s , rather than ξ or η , solves the grid scaling problem. Along the $\eta = \text{constant}$ grid line, the first derivative of a property f with respect to s is $f_s = f_\xi \xi_s$. Differentiating again with respect to s produces:

$$f_{ss} = (f_\xi \xi_s)_s \quad (\text{F.9})$$

This expression can be further manipulated to produce:

$$f_{ss} = (f_\xi)_s \xi_s + f_\xi (\xi_s)_s \quad (\text{F.10})$$

$$f_{ss} = f_{\xi\xi} (\xi_s)^2 + f_\xi \xi_{ss} \quad (\text{F.11})$$

As described in Appendix E, a serious effort is made to insure that grid cell sizes and shapes change only gradually on the physical grid. Therefore, $\xi_{ss} \approx 0$. Assuming this is exactly the case, equation (F.11) becomes:

$$f_{ss} = f_{\xi\xi} (\xi_s)^2 \quad (\text{F.12})$$

Using finite differences, $\xi_s \approx \Delta\xi/\Delta s$. Remembering that s is a function of x and y and that $\Delta\xi = 1.0$, equation (F.12) becomes:

$$\xi_s = \frac{1}{\left[\frac{(x_{i+1,j} - x_{i-1,j})^2}{4} + \frac{(y_{i+1,j} - y_{i-1,j})^2}{4} \right]^{\frac{1}{2}}} \quad (\text{F.13})$$

and

$$(\xi_s)^2 = \frac{4}{(x_{i+1,j} - x_{i-1,j})^2 + (y_{i+1,j} - y_{i-1,j})^2} \quad (\text{F.14})$$

This term is the scaling factor used to eliminate unwanted grid dependent scaling for dissipative derivatives with respect to the ξ direction. Therefore:

$$\text{sf}\xi_2 = \frac{4}{(x_{i+1,j} - x_{i-1,j})^2 + (y_{i+1,j} - y_{i-1,j})^2} \quad (\text{F.15})$$

Similarly, for dissipative derivatives with respect to η :

$$\text{sf}\eta_2 = \frac{4}{(x_{i,j+1} - x_{i,j-1})^2 + (y_{i,j+1} - y_{i,j-1})^2} \quad (\text{F.16})$$

The cascade grids used in the present investigation do not change with time. Also, $\text{sf}\xi_2$ and $\text{sf}\eta_2$ are functions only of grid point locations. Therefore, $\text{sf}\xi_2$ and $\text{sf}\eta_2$ are calculated once and stored, prior to the first time step. These stored values are then used at every time step.

VITA

VITA

Richard McCrea Moore

As the member of a military family he spent the first 17 years of his life at a variety of locations in the United States and on the Japanese island of Okinawa. In 1977 he graduated from the United States Air Force Academy with a Bachelor of Science degree in Aeronautical Engineering. As a lieutenant he was an Advanced Guided Weapons Test Engineer at the Air Force Armament Division at Eglin Air Force Base, Florida. At the Armament Division he tested air-to-air missiles, and remotely piloted target drones. From 1981 to 1982 Captain Moore attended the Air Force Institute of Technology where he graduated with a Master of Science degree in Aeronautical Engineering. In 1983 he returned to the United States Air Force Academy as an Instructor in the Department of Aeronautics where he taught aeronautics, thermodynamics and propulsion. After two years of teaching, the Department of Aeronautics sponsored his Ph.D. studies at Purdue University. After graduating from Purdue University as a Doctor of Philosophy, Major Moore will return to the Air Force Academy as an Associate Professor and the head of the propulsion group.